



Virtualização de processos na Indústria 4.0

Mestrado em Engenharia Eletrotécnica

Tiago Miguel dos Santos Neves

Leiria, novembro de 2020



Virtualização de processos na Indústria 4.0

Mestrado em Engenharia Eletrotécnica

Tiago Miguel dos Santos Neves

Projeto realizado sob a orientação do Professor Doutor Eliseu Ribeiro e do Professor
Doutor Luís Perdigoto

Leiria, novembro de 2020

Originalidade e Direitos de Autor

O presente projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para a elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de Mestrado em Engenharia Eletrotécnica, no ano letivo 2019/2020, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

Agradecimentos

Ao elaborar este projeto, não poderia deixar de agradecer a todas as pessoas com quem troquei conhecimentos e contribuíram para que o projeto tivesse um melhor desenvolvimento.

Aos orientadores deste projeto, Eliseu Ribeiro e Luís Perdigoto, pelo auxílio nas tarefas a realizar, indo ao encontro do que é o objetivo do projeto. Também pela disponibilização de um posto de trabalho na *Siemens Automation Academy*, com acesso a *software* e *hardware* utilizado neste projeto.

À empresa *CADFLOW*, pela sua disponibilidade em demonstrar o funcionamento básico de *software* da *Siemens*[®] (*NX* e *Mechatronics Concept Designer*).

A todos os colegas de mestrado, com quem tive o prazer de partilhar várias ideias na área da Eletrotécnica e com isso tornar mais completo o desenvolvimento do projeto.

Aos meus familiares, por me apoiarem ao longo do curso de mestrado.

Resumo

Neste projeto, foi desenvolvido um sistema de virtualização de processos na Indústria 4.0, com o objetivo de demonstrar a importância de existir uma análise e monitorização, em processos automáticos na área da indústria.

O conceito *Digital Twin*, está a ser utilizado neste tema de virtualização, pois pode ser definido como uma cópia virtual de um processo físico existente, tendo como objetivo interagir com esse, trocando informação e identificar melhorias ao longo do tempo. Também se define por uma simulação de um modelo real, antes de este ser implementado para prever o seu funcionamento.

Tendo em conta os conteúdos descritos, foi realizado um *Digital Twin* de um *kit* de automação, que se encontra na *Siemens Automation Academy*, na ESTG, e é utilizado para dar formação aos alunos, com a utilização de autómatos. Este *kit* é constituído por uma mesa rotativa de montagem, que disponibiliza peças a um operador, e contém vários sensores e atuadores para o funcionamento automático do processo.

A ideia principal é a de utilizar um *software* de modelação 3D, para efetuar o modelo virtual do *kit*, neste caso o *NX* da *Siemens*[®], que tem incorporado o simulador de movimentos e física através de atuadores e sensores, o *Mechatronics Concept Designer*. Para efetuar a programação do *Digital Twin*, é utilizado o *TIA Portal* da *Siemens*[®]. Neste *software*, é possível existir uma interligação, entre eles, através da criação de um autómato virtual, com o *PLCSIM*, criando um sistema denominado por *Software-in-the-loop*. Também é possível interligar o *Digital Twin* com um autómato real, através de um protocolo de comunicação, neste caso designa-se pelo sistema *Hardware-in-the-loop*.

Este projeto, ajuda a perceber como é possível antecipar e otimizar, virtualmente, todos os componentes de um processo físico industrial, antes de este existir, reduzindo riscos, custos e ajustes no chão de fábrica, quando o sistema for instalado. Durante a sua utilização, o espaço virtual, também pode intervir, de maneira a tornar o processo real mais eficiente, informando os envolvidos do processo em tempo real, do funcionamento correto deste.

Abstract

In this project, a processes virtualization system in Industry 4.0 was developed, with the aim of demonstrate the importance of having an analysis and monitoring, in industry area automatic processes.

The Digital Twin concept, is being used in this virtualization theme, as it can be defined as a virtual copy of an existing physical process, aiming to interact with it, exchanging information and identifying improvements over time. It is also defined by a simulation of a real model, before it is implemented to predict its operation.

Considering the contents described, a Digital Twin of an automation kit was made, which is at the Siemens Automation Academy, at ESTG, and is used to train students with the use of PLCs. This kit consists of a rotating assembly table, which provides parts to an operator, and contains several sensors and actuators for automatic operation of the process.

The main idea is to use 3D modulation software, to make the virtual model of the kit, in this case the NX from Siemens®, which has incorporated the motion and physics simulator through actuators and sensors, the Mechatronics Concept Designer. To program the Digital Twin, TIA Portal from Siemens® is used. In these software, it is possible to have an interconnection, between them, through the creation of a virtual PLC, with PLCSIM, creating a system called Software-in-the-loop. It is also possible to connect the Digital Twin with a real PLC, through a communication protocol, in this case it is called the Hardware-in-the-loop system.

This project helps to understand how it is possible to anticipate and optimize, virtually, all the components of an industrial physical process, before it exists, reducing risks, costs and adjustments on the shop floor, when the system is installed. During its use, the virtual space can also intervene, in order to make the real process more efficient, informing the parties of the process value in real time, of its correct functioning.

Índice

Originalidade e Direitos de Autor	iii
Agradecimentos	iv
Resumo.....	v
Abstract.....	vii
Lista de Figuras	x
Lista de Tabelas	xiv
Lista de siglas e acrónimos	xv
1. Introdução.....	2
2. Virtualização de processos.....	6
2.1 Indústria 4.0	6
2.2 Virtualização na indústria	11
2.3 Digital Twin	15
2.4 Virtual Commisioning.....	20
2.5 Software de modelação e programação	23
2.5.1 Programação PLC	24
2.5.2 Criação de modelos virtuais	25
2.5.3 Exemplos de <i>software de simulação</i>	25
2.6 Métodos de interligação e simulação	29
2.6.1 Protocolos de comunicação	30
2.6.2 Exemplos de utilização do protocolo <i>OPC UA</i>	31
3. Desenvolvimento	36
3.1 Processo físico	36
3.1.1 Descrição do processo físico	37
3.2 Modelo virtual.....	39
3.2.1 <i>Siemens NX</i>	40
3.2.2 <i>Mechatronics Concept Designer</i>	41
3.2.3 Operações no <i>MCD</i>	47
3.2.4 Criação de sinais e instruções	54
3.3 Interligação com o Digital Twin	60
3.3.1 Programação do <i>Digital Twin</i>	61

3.3.2	<i>PLCSIM</i>	66
3.3.3	<i>TCP/IP</i>	70
3.3.4	<i>Profinet</i>	74
3.3.5	<i>OPC DA</i>	78
3.3.6	<i>OPC UA</i>	83
4.	Resultados e problemas	86
4.1	Versões de <i>software</i> e <i>firmware</i>	86
4.2	Resultados e problemas na simulação	87
5.	Conclusões	90
	Referências Bibliográficas	92
	Anexos	100

Lista de Figuras

Figura 1 – Revoluções Industriais [6]	6
Figura 2 – Conexão digital entre processos na Indústria 4.0	7
Figura 3 – Sistema de <i>HMI</i> com controlo de monitorização por operador [27].....	13
Figura 4 – Exemplo de sistema de <i>SCADA</i> com vários dispositivos ligados em rede.....	14
Figura 5 – Representação de um <i>Digital Twin</i> de um processo de produção industrial [35].	16
Figura 6 – <i>Layout</i> de indústria do cartão canelado com linha de produção automática [24]	18
Figura 7 – <i>Digital Twin</i> num processo físico existente [40]	19
Figura 8 – Processo de montagem de cilindros pneumáticos [40]	19
Figura 9 – Simulação <i>Software-in-the-loop</i> de uma linha de produção.....	21
Figura 10 - Simulação <i>Hardware-in-the-loop</i> de um motor e o seu controlo elétrico.....	22
Figura 11 – Interface de comunicação na simulação <i>HIL</i> de uma linha de produção industrial [46]	23
Figura 12 – Linha de produção com <i>Factory I/O</i> [58]	26
Figura 13 – Linha de montagem com <i>RobotStudio</i> [60].....	27
Figura 14 – Sistema industrial com <i>Emulate3D</i> [62]	27
Figura 15 - Sistema automático com <i>AGV</i> no <i>Delmia</i> [64]	28
Figura 16 - Projeto de <i>Digital Twin</i> no <i>Siemens NX</i> [44].....	28
Figura 17 – Linha de montagem robótica com <i>OPC UA</i> [74]	32
Figura 18 – Plataforma <i>web</i> com <i>OPC UA</i> [75]	33
Figura 19 – Processo laboratorial com <i>Automation ML</i> e <i>OPC UA</i> [76].....	33
Figura 20 – Sistema de monitorização com <i>MATLAB®</i> e <i>OPC UA</i> [77]	34
Figura 21 – <i>Labview</i> e <i>OPC UA Client</i> numa máquina agrícola [78]	34
Figura 22 – Mesa de montagem rotativa.....	37
Figura 23 – Descrição do painel de comandos da mesa de montagem	38
Figura 24 – Base do comutador de peças do modelo virtual e do processo físico	40
Figura 25 – Modelo digital da mesa de montagem rotativa	40
Figura 26 – Modelação da base da mesa de montagem	41
Figura 27 – Montagem do modelo virtual da mesa de montagem.....	42

Figura 28 – Restrições mecânicas entre a base de peças e a base do modelo virtual.....	43
Figura 29 – Criação de um corpo rígido da base do modelo virtual.....	43
Figura 30 – Criação de corpos de colisão do modelo virtual	44
Figura 31 – Criação de junta fixa da codificação das posições com a roda como base	45
Figura 32 – Criação de junta deslizante do comutador de peças com a base azul como base	45
Figura 33 – Criação de junta articulada da roda.....	46
Figura 34 – Definição de controlo de posição do comutador.....	46
Figura 35 – Operação de posição da roda.....	47
Figura 36 – Operação de posição do comutador	48
Figura 37 – Problema e melhoria nos corpos de colisão.....	49
Figura 38 – Processamento do computador na simulação e erro <i>NX MCD</i>	50
Figura 39 – Avaliação de performance dos gráficos no <i>NX MCD</i>	50
Figura 40 – Criação de fonte de objetos	51
Figura 41 – Otimização das propriedades físicas das peças.....	51
Figura 42 – Sequência de operações no <i>NX MCD</i>	52
Figura 43 – Criação de sensores de colisão e de componentes adicionais	53
Figura 44 – Operação atingir posição 3 com condições de sensores	54
Figura 45 – Criação de sinais de posição e do comutador	55
Figura 46 – Criação de sinais para deteção do tipo de peça.....	55
Figura 47 – Instrução de controlo de velocidade da roda	57
Figura 48 – Instrução de controlo de posição do comutador	58
Figura 49 – Instrução de controlo de velocidade do comutador.....	58
Figura 50 – Instrução de início da simulação.....	59
Figura 51 – Criação do atuador de mudança de cor da peça inferior.....	59
Figura 52 – Instrução de alarme de cor errada	60
Figura 53 – <i>PLC S7-1200</i> utilizado no projeto	61
Figura 54 – <i>PLC S7-1500</i> utilizado no projeto	61
Figura 55 – <i>GRAFCET</i> de programação do <i>DT</i>	63
Figura 56 – Blocos de programação do <i>TIA Portal</i>	64
Figura 57 – <i>DB</i> das variáveis do <i>DT</i>	65

Figura 58 – Arranque do <i>PLCSIM</i>	66
Figura 59 – Download para o <i>PLC</i> no <i>TIA Portal</i>	66
Figura 60 – Acesso a configuração de interligação externa no <i>NX MCD</i>	67
Figura 61 – Configuração da interligação <i>PLCSIM</i> no <i>NX MCD</i>	67
Figura 62 – Correspondência de sinais com <i>PLCSIM</i> no <i>NX MCD</i>	68
Figura 63 – Funcionamento interligação <i>PLCSIM</i>	69
Figura 64 – Bloco <i>TSEND_C</i> no <i>TIA Portal</i>	70
Figura 65 – Configuração do bloco <i>TSEND_C</i> no <i>TIA Portal</i>	71
Figura 66 – Bloco <i>TRCV_C</i> no <i>TIA Portal</i>	71
Figura 67 – Configuração do bloco <i>TRCV_C</i> no <i>TIA Portal</i>	72
Figura 68 – Programação adicional para envio e recepção de sinais	72
Figura 69 – <i>DBs</i> para receber e enviar sinais com o <i>NX MCD</i> via <i>TCP/IP</i>	73
Figura 70 – Configuração da interligação <i>TCP</i> no <i>NX MCD</i>	74
Figura 71 – Criação de <i>PC station</i> e módulos em <i>Profinet</i> no <i>TIA Portal</i>	74
Figura 72 – Rede <i>Profinet</i> entre <i>PLC 1500</i> e uma <i>PC station</i> no <i>TIA Portal</i>	75
Figura 73 – Configuração de <i>PLC</i> para <i>IO device</i> no <i>TIA Portal</i>	75
Figura 74 – Variáveis de entrada e saída do <i>PLC</i>	76
Figura 75 – Configuração do <i>Station Configuration Editor</i> com aplicação.....	76
Figura 76 – Configuração de interligação <i>Profinet</i> no <i>NX MCD</i>	77
Figura 77 – Conexão inicial com <i>Profinet</i> no <i>NX MCD</i>	77
Figura 78 - Criação de <i>PC station</i> e módulos em <i>OPC DA</i> no <i>TIA Portal</i>	78
Figura 79 – Criação da <i>S7 Connection</i> para o <i>OPC DA</i> no <i>TIA Portal</i>	79
Figura 80 – Programação adicional com <i>OPC DA</i> no <i>TIA Portal</i>	79
Figura 81 - Configuração do <i>Station Configuration Editor</i> com <i>OPC Server</i>	80
Figura 82 – Conexão com o <i>Server OPCDA</i> no <i>OPC Scout</i>	80
Figura 83 – Acesso a variáveis do <i>OPC Server</i> no <i>OPC Scout</i>	81
Figura 84 - Criação de variáveis no <i>OPC Scout</i>	81
Figura 85 – Ligação com o <i>OPC DA Server</i> no <i>NX MCD</i>	81
Figura 86 – Seleção de variáveis a interligar no <i>NX MCD</i> com <i>OPC DA</i>	82
Figura 87 – Variáveis booleanas com <i>OPC DA</i> no <i>NX MCD</i>	82

Figura 88 – Configuração do <i>PLC</i> com <i>OPC UA</i> no <i>TIA Portal</i>	83
Figura 89 – Segurança do Server <i>OPC UA</i> no <i>TIA Portal</i>	83
Figura 90 – Ligação com Server <i>OPC UA</i> no <i>NX MCD</i>	84
Figura 91 – Seleção de variáveis a interligar no <i>NX MCD</i> com <i>OPC UA</i>	84
Figura 92 – Problema no <i>kit</i> real de colocação de peça no pino.....	88

Lista de Tabelas

Tabela 1 – Codificação dos sensores de posição	37
Tabela 2 – Sequência de funcionamento do <i>DT</i>	47
Tabela 3 – Lista de sinais e correspondência.....	56
Tabela 4 – Lista de versões compatíveis entre <i>software</i> e <i>firmware</i>	86

Lista de siglas e acrónimos

<i>AGV</i>	<i>Automated guided vehicle</i>
<i>CAN</i>	<i>Controller Area Network</i>
<i>CNC</i>	<i>Computer Numeric Control</i>
<i>DNC</i>	<i>Direct numerical control</i>
<i>DT</i>	<i>Digital Twin</i>
<i>DTSF</i>	<i>Digital Twin Shop-Floor</i>
<i>ESTG</i>	<i>Escola Superior de Tecnologia e Gestão</i>
<i>GPS</i>	<i>Global Positioning System</i>
<i>HIL</i>	<i>Hardware-in-the-loop</i>
<i>HMI</i>	<i>Human Machine Interface</i>
<i>IO device</i>	<i>Dispositivos de entradas e saídas</i>
<i>IoT</i>	<i>Internet of Things</i>
<i>IT</i>	<i>Information Technology</i>
<i>KPI</i>	<i>Key Performance Indicators</i>
<i>MCD</i>	<i>Mechatronics Concept Designer</i>
<i>OEE</i>	<i>Overall Equipment Effectiveness</i>
<i>OPC</i>	<i>Open Platform Communication</i>
<i>OPC DA</i>	<i>OPC Data Access</i>
<i>OPC UA</i>	<i>OPC Unified Architecture</i>
<i>PLC</i>	<i>Programmable Logic Controller - Autómato</i>
<i>PLCSIM</i>	<i>Simulador de controlador virtual</i>
<i>PLM</i>	<i>Product Lifecycle Management</i>
<i>Profibus</i>	<i>Process Field Bus</i>
<i>Profinet</i>	<i>Process Field Net</i>
<i>SCADA</i>	<i>Supervisory Control and Data Acquisition</i>
<i>Siemens NX</i>	<i>Software de modelação 3D da Siemens®</i>
<i>SIL</i>	<i>Software-in-the-loop</i>
<i>TCP/IP</i>	<i>Transmission Control Protocol/Internet Protocol</i>
<i>TIA Portal</i>	<i>Totally Integrated Automation Portal</i>
<i>XML</i>	<i>Extensible Markup Language</i>

VC

Virtual Commissioning

1. Introdução

Com o aumento da utilização de plataformas digitais por parte da população, encontramos-nos no meio de um ambiente tecnológico, com uma significativa transformação, da forma em como os produtos do nosso dia a dia são produzidos graças à digitalização da indústria. Esta transição é tão ambiciosa que se tornou na chamada 4ª revolução industrial, onde esta tem como base a utilização de elementos, do que sucedeu da 3ª revolução, com a adoção de computadores, automação inteligente e otimização de processos [1]. A 4ª revolução industrial deu origem ao conceito de Indústria 4.0, que consiste na conexão digital entre todos os envolventes do valor de um processo, criando uma rede inteligente entre processos e produtos, através de tecnologias de informação e comunicação [2].

No tema da Indústria 4.0, a virtualização de processos é uma importante tecnologia que se pode aplicar, dando origem aos sistemas Ciber-Físicos. A virtualização consiste na realização de uma interface digital, que interage com um conjunto de dispositivos, permitindo obter e processar informação. Nesta aplicação, pode ser utilizado o conceito de *Digital Twin (DT)*, que é constituído por um processo físico que contém máquinas, produtos e operadores, e por um espaço virtual, onde contém todos os elementos do processo físico no formato digital [3]. Através de um protocolo de comunicação, é possível interligar o sistema físico com o virtual, para aquisição de dados e identificação de melhorias, no processo global, ao longo do tempo. Com esta interligação, é possível obter dados relevantes de um sistema de produção de uma empresa e efetuar planeamentos de trabalho ou de manutenção para otimizar o funcionamento dessa empresa [4].

Outra aplicação importante do *DT*, é na simulação de um processo real, antes de ser implementado, para identificar como se vai comportar a nível mecânico, elétrico e tempos de ciclo. Com esta aplicação, existe a possibilidade de poupar tempo e dinheiro, ao encontrar problemas no espaço virtual, antes de um sistema ser finalizado. Enquanto um projeto estiver a ser implementado, também é possível, a qualquer altura, configurá-lo para uma determinada tarefa diferente ou adicionar alguma componente, tendo em conta futuros cenários, e verificar o seu comportamento com essa mudança [5].

No início do mestrado, no ano letivo 2017/2018, foi lançado o desafio pelos orientadores, ao autor deste projeto, de explorar *software* de modelação 3D e a sua conexão com autómatos da marca *Siemens®*. Na ESTG, existe a *Siemens Automation Academy*, que contém vários

produtos da *Siemens*[®], a nível de *hardware* e *software*. Ao aceitar esse desafio, definiu-se o objetivo do projeto, que é o de elaborar um sistema de virtualização de processos na Indústria 4.0, através da implementação de um *DT*. Foi criado um modelo digital de um processo físico, existente na academia, com o objetivo de verificar a interação entre os dois espaços (virtual e físico) podendo obter melhorias e observações relevantes no funcionamento deste.

Inicialmente no projeto, verificou-se como funciona o processo, que é uma mesa de montagem rotativa. É constituída por uma roda com várias posições, em que cada uma contém um pino, e por um comutador que se desloca, fazendo cair dois tipos de peças distintas. Para além disto, também contém sensores para determinar as posições da roda, o tipo e presença das peças.

Para realizar o modelo digital do processo físico, foi necessário pesquisar sobre o *software* de modelação 3D, neste caso o *Siemens NX*, verificando as suas potencialidades a nível de operações e da realização de componentes com diferentes tipos de geometria. Ao mesmo tempo, verificar o comportamento mecânico do processo e efetuar as medições necessárias de todos os componentes físicos, para uma melhor implementação deste modelo. Após efetuar o desenho 3D do processo, iniciou-se a simulação de movimentos, com o *MCD*, que é uma aplicação do *Siemens NX*. Nesta aplicação, definiu-se uma sequência de montagem das peças, e, consoante isso, verificou-se como executar movimentos, tendo em conta a capacidade do *software* em termos de comportamentos de colisão de objetos e de processamento.

Com as questões físicas e de movimentação alinhavadas, procedeu-se à interligação entre o *MCD* e o autómato. Nesta tarefa, é utilizado o *TIA Portal* que é um programa que permite executar tarefas de automação e vai ser responsável pela lógica de programação. Este *software* tem potencial na área de automação permitindo também configurar redes de comunicação e controlo de supervisão. Existem vários tipos de interligações possíveis, que tanto o *TIA Portal* como o *MCD* possibilitam, mas as que foram estudadas são o *PLCSIM*, *TCP/IP*, *Profinet*, *OPC DA* e *OPC UA*. O *PLCSIM* cria um controlador virtual simulando um autómato real da *Siemens*[®], sem a necessidade de existir qualquer tipo de *hardware*. Isto é ideal para criar um sistema *Software-in-the-loop (SIL)*, que simula o desempenho do *software* a nível de programação e do ambiente de modelação do sistema virtual. As outras interligações, são protocolos de comunicação utilizados na indústria com autómatos reais. Podem ser usados para criar um sistema *Hardware-in-the-loop (HIL)*, no sentido de existir uma conexão real entre o modelo virtual e o autómato, para avaliar o desempenho do processo real.

Neste relatório de projeto, a estrutura está dividida em vários capítulos, começando por uma parte teórica sobre o tema de virtualização e suas aplicações, para melhor perceber o significado deste conceito e da sua importância no âmbito da Indústria 4.0. De seguida, é explicado o potencial do *software* utilizado, bem como os protocolos de comunicação. Após isso, é descrita a parte técnica do projeto, sobre o seu desenvolvimento, resultados e problemas encontrados. Por fim, serão apresentadas as conclusões obtidas, ao desenvolver este projeto e o possível trabalho futuro considerando o tema do *DT*.

Por fim, é importante divulgar, que este trabalho participou no prémio geração digital !19, realizado pela *Siemens*[®] Portugal, tendo ficado em primeiro lugar.

2. Virtualização de processos

Este capítulo apresenta os conceitos teóricos de virtualização e as suas aplicações, no conceito da Indústria 4.0. Os vários tipos de virtualização, *software*, *hardware* e protocolos de comunicação, vão ser descritos para mostrar a sua potencialidade na área da indústria e investigação.

2.1 Indústria 4.0

A área da indústria tem vindo a ter, ao longo dos anos, uma crescente evolução tecnológica e de recursos que influencia a forma em como nos comportamos a nível profissional e pessoal, criando uma nova estrutura social-económica. A sua evolução foi caracterizada por várias revoluções, cada uma marcada por um acontecimento histórico associado, como se vê na Figura 1.

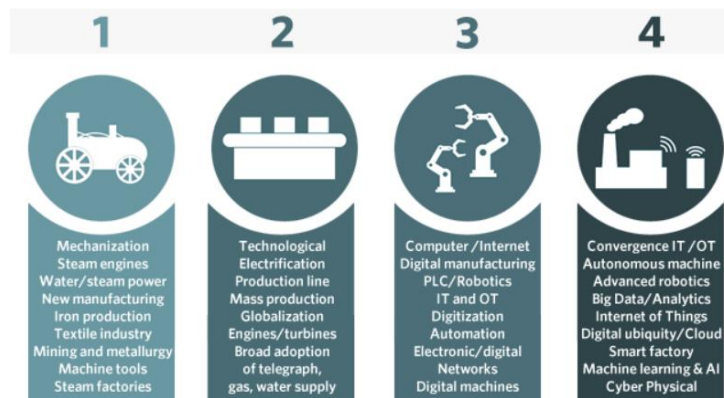


Figura 1 – Revoluções Industriais [6]

A 1ª revolução industrial, consistiu na utilização de energia mecânica, através de motores a vapor de água, ao invés de utilizar o trabalho artesanal de operadores, em produtos comerciais. Com a elaboração de novos métodos de trabalho, melhoria nos processos, e novas técnicas descobertas, surgiu a 2ª revolução industrial, com produção em massa e linhas de montagem através do auxílio da eletricidade, que teve impacto neste acontecimento. A 3ª revolução surgiu com o aparecimento de *Programmable Logic Controllers (PLCs)*, robótica e computadores, tornando os processos da indústria automáticos e autónomos [6]. Nos dias de hoje, estamos perante um crescente avanço tecnológico de vários elementos como [7]:

- *Internet of Things*, definida pela troca de informação entre pessoas, produtos e processos, através da Internet, podendo obter informações importantes que dão origem à melhoria de serviços;

- *Big Data*, um grande conjunto de dados de várias fontes que são formatados numa determinada estrutura *standard*, tornando eficiente a tomada de decisões;
- *Cloud Computing*, armazenamento de dados na nuvem através de servidores e cujo acesso é via Internet, possibilitando um menor risco de perder informação e um acesso flexível a esta;
- Sistemas Ciber-Físicos, que são um conjunto de dispositivos físicos que comunicam com uma plataforma digital, por meio de uma rede de comunicação, permitindo a redução de falhas e a otimização de processos.

Por via destas tecnologias, surgiu a 4ª revolução industrial, também denominada por Indústria 4.0, que tem como objetivo tornar a indústria mais eficiente, produtiva e a criação de novos produtos. A Indústria 4.0, surge da evolução crescente das tecnologias de informação e comunicação. A aplicação de automação inteligente de sistemas Ciber-Físicos, com redes de comunicação avançada, interligadas com a Internet (*IoT*), permite um controlo descentralizado, atingindo a característica da interoperabilidade. A Indústria 4.0, tem o potencial de reorganizar os processos de produção industriais, no sentido de os tornar mais autónomos, organizados e flexíveis na qualidade e personalização de produtos [8].

• Tecnologias da Indústria 4.0

Novos conceitos foram surgindo, com esta evolução tecnológica e também inseridos nesta nova revolução industrial. O objetivo é o de criar uma conexão digital entre vários processos, como se pode verificar na Figura 2. Alguns desses conceitos tecnológicos, vão ser apresentados de seguida.



Figura 2 – Conexão digital entre processos na Indústria 4.0

● Impressão 3D

Por exemplo a impressão 3D, é um processo que executa a elaboração de produtos, com uma determinada geometria, utilizando material polimérico. É possível criar um produto através de modelação 3D, que pode ser utilizado em protótipos ou em *spare parts* (peças de reposição). A impressão 3D, insere-se na Indústria 4.0 como sendo um potencial na criação produtos personalizados, tendo em conta os requisitos do utilizador. É claro que para uma produção em massa de certos produtos, a impressão 3D ainda necessita de uma evolução, mas por exemplo para efetuar uma peça mecânica específica na substituição de um equipamento, pode ser o ideal. Esta tecnologia está a ser utilizada por universidades para formar estudantes e os preparar para as necessidades do mundo da Indústria 4.0 [9].

● Realidade aumentada e realidade virtual

A realidade aumentada e a realidade virtual também são novos conceitos. A primeira executa a compilação de imagens sobrepostas numa realidade existente, através de dispositivos móveis. Esta torna a perceção da realidade, mais significativa, devido à interação que existe com certos objetos no mundo real. Os objetos podem ter um código ou um manual de operação, para um operador poder identificar como deve proceder numa determinada tarefa. Por sua vez, a realidade virtual, através de um ecrã usado na cabeça ao nível dos olhos (*Head-Mounted Display*), executa a simulação artificial, que recria um ambiente real. Esta pode ser usada na execução de tarefas no mundo virtual para aplicação posterior no mundo real. Ou seja, pode ser usada na realização de formações de operadores ou em trabalhos mais exigentes, auxiliando a prevenir algum tipo de acidente ou de aumento de eficácia no trabalho, por exemplo [10].

● Inteligência artificial

O conceito da inteligência artificial, é definido pela execução de tarefas para resolver problemas complexos, de forma mais rápida, segura e otimizada que o ser humano. O *Machine Learning*, é uma das vertentes para atingir a inteligência artificial tendo em conta que é possível aplicar algoritmos, onde realizam uma análise e estruturação de dados (*Big Data*). Numa empresa, através de informação obtida dos equipamentos e do pretendido de um cliente, de forma automática, pode ser possível avaliar qual a melhor forma de executar o planeamento de produção de vários produtos, ou até de executar a manutenção preditiva e corretiva, tendo em conta o funcionamento dos equipamentos e dos seus componentes [11][12].

● *Cloud Computing*

O *Cloud Computing* permite o armazenamento de dados, via internet, onde várias entidades podem ter acesso, garantindo uma boa troca de informação. É uma tecnologia que existe há algum tempo, mas tem potencial no conceito da Indústria 4.0, pois devido ao grande número de dados a armazenar, numa indústria, é possível fazê-lo através da Internet. Ao retirar espaço de armazenamento de dados internamente, não é necessário um grande poder computacional, nos computadores das empresas. Esta tecnologia tem vindo a ser implementada na componente digital de sistemas Ciber-Físicos, no sentido de garantir uma escalabilidade do armazenamento de dados, capacidade de processamento e comunicação entre vários domínios [13].

● *Internet of Things (IoT)*

A *Internet of Things* (*Internet* das coisas) consiste em objetos inseridos num equipamento com tecnologias integradas, que permitem obter, organizar e transmitir dados, para um determinado objetivo. A *IoT*, é uma das tecnologias existentes hoje em dia, e que terá impacto na Indústria 4.0. A tecnologia em questão, pode ser aplicada em sistemas Ciber-Físicos, que são constituídos por sensores e atuadores, e por uma rede de comunicação, que pode estar conectada à Internet, sendo uma característica da *IoT* [14].

● **Aplicações da Indústria 4.0**

Um dos benefícios esperados com a implementação da Indústria 4.0, é o aumento da eficiência dos processos industriais, no entanto nem todas as empresas estão realmente a implementar as tecnologias associadas, portanto é um processo que está ainda em adaptação e evolução à medida do tempo. Este conteúdo é recente, e é normal que ainda sejam necessários mais estudos sobre o seu impacto nas empresas e no que pode ajudar na sociedade, a nível financeiro, organização e segurança [15]. Tendo em conta a ideia das empresas, um novo conceito surgiu, proveniente da Indústria 4.0, que é a *Smart Factory* (Fábrica Inteligente). O objetivo é de atingir a comunicação entre todas as máquinas, produtos e operadores, para obter dados e processá-los, de maneira a melhorar a competitividade, eficiência, produtividade, e a criação de novos produtos de maior qualidade e personalização [16].

Um exemplo de aplicação, encontra-se numa empresa fabricante de engrenagens que adotou um sistema Ciber-Físico de logística. Este sistema foi aplicado num processo que é constituído por várias máquinas, que fabricam um determinado produto, e em cada máquina existe uma estação de entrega de produto final. Periodicamente, um operador recolhe esse produto e entrega componentes necessários para a realização da fabricação. O sistema Ciber-Físico tem como objetivo, melhorar a eficiência logística do funcionamento de recolha do produto, onde as linhas de produção têm muitas variações e não se encontram sincronizadas. Este procedimento permite obter mais informação, de forma digital, e através de uma rede de comunicação, sobre a produção de cada máquina. O operador, consegue efetuar uma melhor análise, no sentido de quando deve abastecer ou recolher produtos da máquina. Para a empresa, isto traz mais valias no sentido de redução de *stocks* e trabalho mais organizado e flexível dos operadores [17].

No ramo automóvel, também foi implementado um projeto de digitalização dedicado a linhas de produção de uma fábrica que produz jantes. O objetivo do projeto, é de recolher informação sobre as linhas de produção, efetuando ajustes necessários para melhorar a sua eficiência. Para tal, é usado o cálculo do *Overall Equipment Effectiveness (OEE)*, que é definido pela forma como se usa o tempo, eficientemente, para fabricar produtos de qualidade (valor da produção). O cálculo deste indicador, necessita de várias informações, disponibilizadas por um sistema de monitorização aplicado na fábrica, que indica tempos de ciclos, falhas, quantidade de produção etc. Com esta informação, é possível avaliar o desempenho das linhas de produção, em tempo real, de forma a efetuar um melhor planeamento e aplicar melhoria contínua. Este sistema de monitorização aplicado, também permite uma maior flexibilidade de disponibilização de informação, entre os vários setores da empresa [18].

Outro projeto foi implementado numa empresa na área do aço, onde se pretende reduzir os custos de produção e aumentar a competitividade do mercado. O objetivo do projeto, é o de verificar a precisão de vários componentes, como sensores, tecnologias de rádio e sistemas de localização, implementados em veículos internos da fábrica, de modo a otimizar o seu desempenho. Este projeto é constituído por um sistema de monitorização e supervisão, onde se encontram implementadas tecnologias como o *Cloud Computing* e o *IoT*, que possibilitam várias operações aos operadores e gestores. A aplicação elaborada permite localizar operadores, observar o estado dos veículos, análise de alarmes, informar sobre ferramentas necessárias para realizar tarefas, etc. Tendo em conta a implementação da digitalização neste projeto, esta permitiu o aumento da eficiência energética e material [19].

De acordo com as aplicações mencionadas, a Indústria 4.0 oferece inovação em processos das empresas, através de plataformas digitais. Empregos vão ser alterados com o avanço tecnológico, ao longo do tempo, visto que diversos sistemas se tornarão autónomos e a possibilidade do seu controlo ser remoto. A parte digital, está cada vez mais integrada, nos processos físicos da indústria [20].

2.2 Virtualização na indústria

Algumas das tecnologias mencionadas no tópico anterior, são caracterizadas por serem virtuais (*IoT*, *Cloud Computing*, *Big Data*), onde são aplicadas em *hardware*, que melhora o seu funcionamento, e é nesse aspeto que a Indústria 4.0 se centra, na era digital da indústria. A virtualização é um processo baseado em *software* ou uma plataforma digital que representa várias componentes como aplicações virtuais, armazenamento, redes de comunicação ou servidores [21].

Com a virtualização introduzida na indústria, a troca de informação entre os diversos equipamentos existentes na unidade fabril e o seu armazenamento na nuvem, um sistema virtual, pode ser capaz de elaborar um conjunto de recursos. Estes recursos podem influenciar diretamente nos processos da empresa, de forma autónoma, ou auxiliar na decisão dos gestores. A digitalização também pode auxiliar na supervisão remota de um sistema automático, sem a necessidade de estarmos próximos deste, pois pode existir sempre um custo associado à deslocação de um técnico para resolver uma determinada anomalia [21][22].

Cada vez mais, a virtualização é implementada nos equipamentos do nosso dia a dia e têm influência no nosso modo de vida, como por exemplo os *smartphones* e as aplicações criadas com várias tendências. Na indústria e nas empresas, a virtualização, está a ser aplicada em projetos a desenvolver, que dão origem a processos mais dinâmicos e tecnológicos. Pode existir uma prevenção de trabalhos a realizar como, manutenção preditiva ou prever o planeamento do processo de produção, acoplando nestes, informação obtida do *Big Data*, *Cloud Computing* e do *IoT*. Devido à necessidade das empresas terem de criar planeamentos de trabalhos, para atingir certos objetivos de produção, pode existir um processo digital, que consiga efetuar toda a gestão, tendo em conta as informações obtidas das várias secções de uma empresa, desde produção, manutenção, qualidade, logística, entre outros. A tendência atual e futura, é a de fazer com que os gestores industriais, procurem outro tipo de *software*, aplicações digitais ou melhorar o sistema de gestão existente, para que sejam dedicados a definir melhor as tarefas a desempenhar em tempo real [21].

Apesar desta componente digital poder ser utilizada na indústria, tem de existir uma avaliação correta do tipo de aplicação, do custo de implementação e do rendimento que esta pode trazer ao ambiente industrial. Outro fator importante, é a garantia de segurança nos dados que são transmitidos entre os equipamentos de uma unidade fabril ligados à internet, pois podem surgir vírus na rede industrial que interferem no funcionamento dos equipamentos, podendo causar danos e limitar os trabalhos de uma fábrica [21].

● Aplicações de virtualização

Em termos de aplicações digitais na indústria, existe *software* de gestão, criado por entidades de desenvolvimento de *software*, atualmente já aplicados em empresas. Esse *software* serve para efetuar a gestão de produção, manutenção, *stocks* de armazéns, encomendas e outros, como por exemplo, o *PHC*, que é utilizado em empresas no Distrito de Leiria e não só [23][24][25]. Existem empresas que conseguem interligar este tipo de *software* entre os vários departamentos de uma fábrica para dar acesso global de todos os dados envolvidos e importantes para as tarefas a executar por cada um. Por exemplo, existe uma empresa multinacional (onde o autor trabalhou), que utiliza um *software* de gestão de *stocks* de peças na manutenção e tem fábricas em Portugal e Espanha. Tanto a fábrica de Portugal como a de Espanha, podem ter acesso ao consumo e *stock* de peças de cada um, no sentido de se auxiliarem caso algum deles tenha uma rutura de *stock* e necessite urgentemente de uma peça em específico [23][26].

Existe também um sistema dedicado aos processos automáticos existentes na indústria, que executa a interface digital com estes, para monitorizar e efetuar o seu controlo, denominado por *HMI* (Figura 3). Este sistema contém um determinado conjunto de operações através de um ecrã tátil e botões em que, normalmente, comunica com um *PLC*, através de um protocolo de comunicação industrial, que é o responsável pelo processamento de todo o sistema automático. Pode ser utilizado para verificar e controlar valores de grandezas como temperaturas, velocidades, pressão, tempos de ciclos, contagens e sinais digitais como fotocélulas, motores, cilindros pneumáticos etc. Na criação de um projeto *HMI*, é necessário elaborar num *software* específico, a programação da visualização gráfica do sistema, os componentes que este vai interagir entre o operador e a máquina e a configuração de conexão com o *PLC*, indicando os endereços necessários para a troca de dados [27].



Figura 3 – Sistema de *HMI* com controlo de monitorização por operador [27]

Um projeto foi desenvolvido, com o intuito de criar um sistema de impressão automática, através de um ecrã, usando um *PLC* e um *HMI*. O projeto tem como propósito elaborar um controlo de um sistema de impressão, existente numa fábrica, tornando o seu funcionamento de forma automática, sem a necessidade de operadores. Com esta implementação, é possível atingir mais velocidade e melhor desempenho da máquina [28].

Outro projeto foi também desenvolvido tendo em conta o equipamento *HMI*, que foi o de implementar uma ferramenta que avalia a forma como os operadores utilizam os *HMIs* numa unidade fabril. Melhorias nos fatores de interação, autonomia e competência dos operadores, através da criação de um novo design, fazem parte do objetivo deste projeto. Este projeto apresentou novas experiências entre os operadores e as máquinas e no futuro é possível integrar esta solução noutros sistemas idênticos [29].

Um sistema idêntico ao *HMI* é o *SCADA*, no entanto este tem mais funcionalidades. Enquanto que o *HMI* é mais dedicado à visualização gráfica do processo automático, para auxiliar os operadores na supervisão desse, o *SCADA*, permite uma maior capacidade de armazenamento de dados e a ligação a uma base de dados, coisa que o *HMI* não executa. O sistema *SCADA* pode ser considerado um computador industrial e é usado para monitorizar processos em maior escala, como por exemplo, um conjunto de vários dispositivos e equipamentos existentes numa empresa, para integrar a informação de forma global do sistema de produção. O *SCADA* auxilia uma empresa a coordenar a sua eficiência, processar informação e minimizar paragens de produção (*Downtimes*) para efetuar planos de trabalho, *KPIs*, melhoria contínua, entre outros. O *HMI* como é um sistema aplicado localmente, apenas a um determinado processo, pode fazer parte de um sistema de *SCADA* (Figura 4) [30].



Figura 4 – Exemplo de sistema de SCADA com vários dispositivos ligados em rede

Um projeto elaborado que envolve o *SCADA* e o conceito da Indústria 4.0, denomina-se por *Secure Plug and Work*. Este projeto permite que as empresas fabricantes de equipamentos ou máquinas, consigam poupar tempo e dinheiro nos arranques dos dispositivos instalados. Essa característica, deve-se ao facto de existirem dados armazenados nos equipamentos, que podem ser importados para monitorização e controlo, de forma direta, ao conectar à rede da fábrica, ao invés de o fazer de forma manual. O *SCADA* entra neste tema como sendo um equipamento que se modifica de forma autónoma, aos novos equipamentos interligados na rede, ou ao ligar-se numa nova rede, onde esteja implementado este projeto [31].

Outro projeto foi efetuado, na criação de um processo de fabricação, no chão de fábrica, com ênfase na Indústria 4.0. O projeto consiste em demonstrar a interligação entre processos automáticos existentes no chão de fábrica, com tecnologia *IoT* implementada, e sistemas de gestão (*IT-Information Technology*). O *SCADA* neste projeto, foi utilizado para monitorizar o sistema Ciber-Físico criado, verificando parâmetros e informação necessária, através da conexão a um *PLC*. Este projeto trouxe mais valias também no facto de ser possível a criação de base de dados (*Big Data*) e o seu armazenamento na *Cloud* com o auxílio do *SCADA* [32].

Existem vários tipos de virtualização que podem ser aplicados na Indústria 4.0, como o facto de cada equipamento estar conectado por uma rede de comunicação a um sistema *SCADA*, incorporando a ligação à Internet e este armazenar informação na *Cloud*, para efetuar um método de otimização mais preciso. Estando todos os equipamentos ligados em rede, como máquinas, linhas de montagem, e outros, pode auxiliar no contacto direto entre esses e os fornecedores e clientes [33].

2.3 *Digital Twin*

O conceito *Digital Twin (DT)* surgiu na apresentação efetuada para a criação de um centro de *Product Lifecycle Management (PLM)*, numa universidade de *Michigan (USA)*, e foi mencionado pelo *Dr. Michael Grieves*, em 2002. Nessa apresentação ele demonstrou vários elementos como um processo físico real, um espaço virtual e a conexão entre esses dois sistemas para a troca de dados [34].

O *DT* pode ser definido como uma cópia virtual de um processo físico existente, que tem como objetivo interagir com este, trocando informação e identificar melhorias (otimizar) ao longo do tempo. Pode ser utilizado para a monitorização de um processo físico, com um design virtual, idêntico ao real. O *DT* permite a obtenção de dados, em tempo real, de linhas de produção de uma empresa, verificar e ter controlo, de forma remota, no decorrer do funcionamento dos processos. Tem a possibilidade de auxiliar operadores, no sentido de otimizar os processos, adaptar certos produtos aos requisitos de mercado e muitas outras vertentes, de maneira a tornar o funcionamento de uma empresa mais eficiente [35][36].

Por outro lado, pode também funcionar como um simulador de um processo real antes de este ser implementado. Permite identificar como se vai comportar, a nível mecânico, elétrico, pneumático, ou tempos de ciclo, sendo possível encontrar problemas no mundo virtual, antes de um processo ou máquina serem finalizados, e assim fazer poupar tempo e dinheiro. Quando se desenvolve processos automáticos ou máquinas, por vezes recorre-se a um *Layout*, para iniciar o processo de desenvolvimento desse produto, e, por sua vez, pode não ser possível observar certos cenários que possam vir a acontecer, sendo que este conceito ajuda nesse aspeto. Enquanto um projeto estiver a ser implementado, o *DT* permite também, a qualquer altura, configurá-lo para tarefas diferentes ou adicionar alguma componente, tendo em conta futuros cenários possíveis e verificar o comportamento do sistema, mais facilmente, de forma digital. Na Figura 5, pode-se verificar uma representação de um *DT* [37].

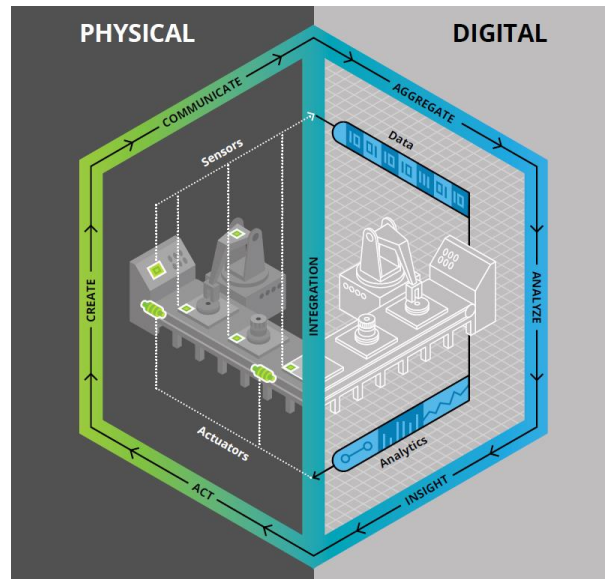


Figura 5 – Representação de um *Digital Twin* de um processo de produção industrial [35].

A ideia principal do *DT*, é de que a informação digital elaborada para a execução de um sistema físico, poderia fazer parte dele próprio. Com a criação de um gêmeo digital, pode se verificar informações necessárias, sobre componentes inseridos no sistema real e fazer parte deste, durante todo o seu tempo de vida. De acordo com o *Dr. Michael Grieves*, o mundo virtual e real, devem estar conectados em 4 fases, a criação, produção, operação e desativação [34].

Na fase de criação, o processo real ainda não existe, sendo então elaborado um *DT* para simular o funcionamento real, em vez de investir na criação de um protótipo. Após a criação, segue-se a parte em que é fabricado o processo real e, ao longo deste, o *DT* pode ser alterado ou atualizado, consoante o decorrer das necessidades das suas configurações. Na fase de operação, é apurado o comportamento do sistema físico verificando as falhas inesperados e aí é aplicada uma ligação entre o sistema real e virtual, para identificar exatamente, o correto funcionamento deste. Por último, chega a fase de desativação, que se foca em guardar informação sobre o comportamento deste processo, para ganhar conhecimento, e na próxima criação de um novo, evitar perder tempo, da forma como todo o processo funciona [34].

Para analisar a importância deste tema do *DT* nas empresas, um novo conceito é criado denominado por *Digital Twin Shop-Floor (DTSF)*. Este conceito, é caracterizado por existir um processo físico, em que, inicialmente, apenas dependia dele próprio composto por operadores e máquinas. Ao adicionar um espaço virtual, passa a existir uma interação onde é gerido todos os controles de produção, tornando o processo mais dinâmico. Atualmente, estão a ser implementadas as tecnologias *IoT*, *Big Data*, *Cloud Computing*, e outros, que vão tornar

a unidade fabril cada vez mais desenvolvida. Possibilitam a obtenção de vários dados internos e externos, através da Internet, como fornecedores e clientes, que em conjunto, dão origem a produtos e equipamentos mais personalizados e de melhor qualidade para os utilizadores. No futuro, com o avanço das tecnologias de informação, o espaço virtual vai ser tão importante como o físico, porque vai ter um impacto progressivo nas tarefas da parte física [38].

● Exemplos de *Digital Twin*

De acordo com a definição de *DT*, muitas aplicações são possíveis em várias áreas como saúde, construção civil, automóvel, e a indústria, que é a área onde se centra maior parte deste tema e deste projeto. Sabe-se que nesta área, são usados muitos componentes de vários tipos, desde elétricos, mecânicos, pneumáticos e hidráulicos, mais concretamente, sensores, motores, electroválvulas, cilindros pneumáticos, etc.

Imagine-se um *DT* aplicado a um motor elétrico, onde este pode ser modelado em 3D (formato digital), para ser idêntico ao seu aspeto físico. O *DT* pode conter todas as dimensões corretas e os seus componentes, e, para além disso, analisar vários parâmetros que se utilizam para controlar esse motor, como posição, velocidade, aceleração, rotação, corrente, binário e outros. Este *DT*, pode servir para prever o desempenho do motor, antes de este ser concebido, e até monitorizá-lo, após ser instalado, de maneira a definir os parâmetros ótimos deste de acordo com a sua função [39].

Um *DT* ser aplicado a um motor é apenas um pequeno exemplo. Empresas na área da indústria, que produzem determinados produtos, estão sujeitas a mudanças com regularidade, no seu chão de fábrica, devido às necessidades de mercado e à sua cadeia de produção. Um *DT*, como já referido, pode ser aplicado a um processo em específico numa empresa para verificar a sua funcionalidade. Na Figura 6, estão representados dois *Layouts* de um sistema físico, obtidos através da empresa onde o autor trabalha atualmente. Este sistema demonstra uma linha de produção automática, na indústria de cartão canelado, onde o *DT* poderia ter sido aplicado, antes da implementação deste processo [24].

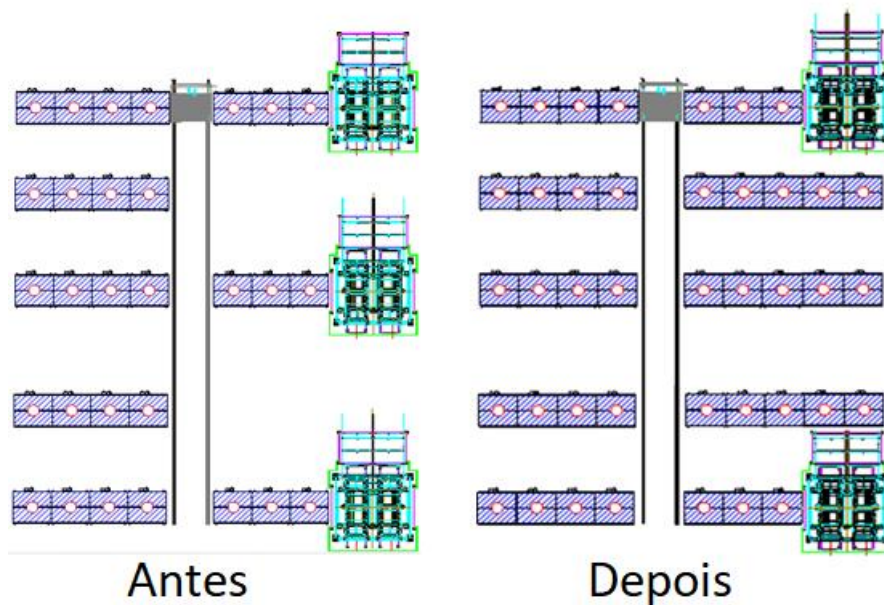


Figura 6 – Layout de indústria do cartão canelado com linha de produção automática [24]

O *Layout* do lado esquerdo da Figura 6, demonstra um processo automático, na indústria do cartão canelado, composto por 5 máquinas, um carro e 3 introdutores. Com o decorrer do tempo, a empresa apercebeu-se que tinha um problema, que era o facto de quando existiam picos de produção de cartão, os transportadores de saída das máquinas ficavam cheios e as máquinas por vezes necessitavam de parar. Após vários estudos internos, optaram por retirar um introdutor de paletes e acrescentar mais transportadores para funcionarem como linhas de buffer, como se vê na Figura 6, do lado direito. Desta forma, foi garantido um processo mais fluído com um melhor seguimento de cargas de cartão. Ao criar um *DT*, aplicado a este processo antes de o implementar, podia servir para efetuar a simulação deste, conseguindo obter a quantidade de produção que se iria realizar. A empresa podia se aperceber logo que tinham de existir alterações para que as máquinas não parassem e causassem custos adicionais, para além do custo de implementação, do primeiro e do segundo processo [24].

Outro tipo de aplicação vantajosa do *DT*, é no aspeto de acompanhar um processo físico ao longo da sua vida, de forma a existir otimização contínua, manutenção dinâmica e processamento de dados. Na Figura 8, verifica-se um processo físico automático para manuseamento de componentes de cilindros pneumáticos. Este projeto foi concebido por estudantes da universidade de tecnologia da Eslováquia, em parceria com algumas empresas na área digital, como a *Siemens*[®] [40].



Figura 8 – Processo de montagem de cilindros pneumáticos [40]

O projeto consiste numa estação de montagem de cilindros pneumáticos, composto por peças, manipuladores, um local de medição e furação, e transportadores. Relativamente à parte digital deste processo, esta foi criada num *software* da Siemens®, que é o *Plant Simulation*. Este *software* permite efetuar uma cópia virtual do processo descrito como se pode verificar na Figura 7 [40].

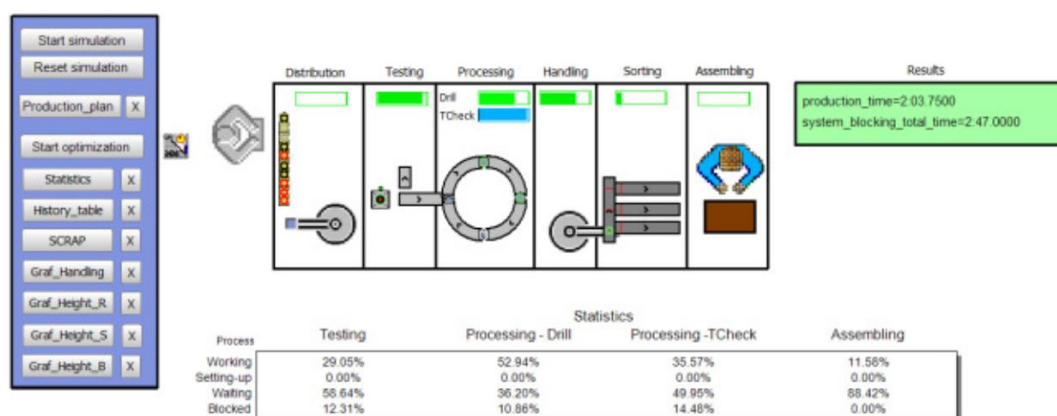


Figura 7 –Digital Twin num processo físico existente [40]

Para colocar em funcionamento o *DT*, utilizou-se os mesmos tempos contabilizados no processo real, fazendo com que este obtenha o tempo de cada tarefa e assim processar dados. Para realmente perceber o impacto que o *DT* pode ter neste processo físico, podem existir certos aspetos, como por exemplo, o comportamento da linha de produção ao alterar determinados parâmetros através do *DT*. Os autores deste projeto decidiram efetuar esse teste, ao forçar um dos manipuladores numa determinada posição ficando este fixo, sem se mover. O *DT* consegue verificar que o processo não está a funcionar dentro do tempo de ciclo definido e pode avisar o operador deste erro e até dar instruções do que deve fazer para o corrigir [40].

2.4 *Virtual Commissioning*

Aquando a elaboração de um processo físico automático na indústria, é necessário recolher as informações importantes do seu correto funcionamento e finalidade, para efetuar o desenho de *Layouts*, esquemas elétricos, programação e depois passar à sua implementação, o que faz com que nestas tarefas, exista por vezes, perda de tempo e dinheiro. O processo ao ser instalado numa unidade fabril, existe uma fase importante na entrega deste, denominado por comissionamento. O comissionamento é definido por existirem entidades competentes, que garantam o correto desempenho do processo, após este ser instalado, elaborando uma série de ações, como verificar os controlos, segurança, formação a operadores, testar alarmes etc. Esta fase é importante e deve ser bem preparada por parte do fabricante e do cliente, pois por vezes é descartada e provoca no futuro anomalias imprevistas ou paragens de produção [41].

O conceito de *DT* pode ser aqui aplicado, pois como representam, na forma digital, um processo físico, também podem ser utilizados na fase de comissionamento. Com esta ferramenta, um novo conceito surge denominado por *Virtual Commissioning* (*VC*). Isto é, em vez de efetuar o comissionamento de um sistema, apenas no mundo físico, é possível criar um *DT* para testar e verificar a simulação do modelo num ambiente virtual. O *VC*, tal como o *DT*, pode auxiliar, de forma digital, a testar o sistema automático, identificação de anomalias, operações e o papel de um novo sistema adicionado a outro já existente. De forma geral, o *VC*, permite informar as características de um processo físico a ser instalado numa fábrica, às pessoas competentes, no sentido de aplicar novas soluções no ambiente fabril e verificar o seu impacto económico, tudo isto antes de ser instalada a solução [42][43].

Este novo conceito, pode ser usado para validar e verificar as capacidades, por exemplo, de uma linha de produção, a nível mecânico e elétrico, através da simulação com ligação a um *PLC*, reduzindo erros e encontrar melhorias. A simulação permite utilizar modelos virtuais de um sistema físico, que têm de corresponder à realidade, e testar o funcionamento deste com a programação de um *PLC* integrado, para depois ser mais fácil e seguro aplicar no mundo real [43][44].

Existem dois tipos de simulação utilizados no VC e também já descritos como trabalho efetuado neste projeto, que são o *Software-in-the-loop (SIL)* e o *Hardware-in-the-loop (HIL)*. O *SIL* significa que não se tem qualquer tipo de *hardware* na simulação, ou seja, os modelos virtuais de um sistema físico e a lógica de programação de um *PLC*, são simulados através de um computador, que executa um controlador virtual (*PLC* virtual), interligado ao modelo digital (ver Figura 9). O *SIL* é utilizado em várias situações, antes da implementação no mundo real, para prever algum tipo de funcionalidade e verificar possíveis alterações, através da correção de elementos mecânicos nos modelos virtuais ou de programação do *PLC* [45].

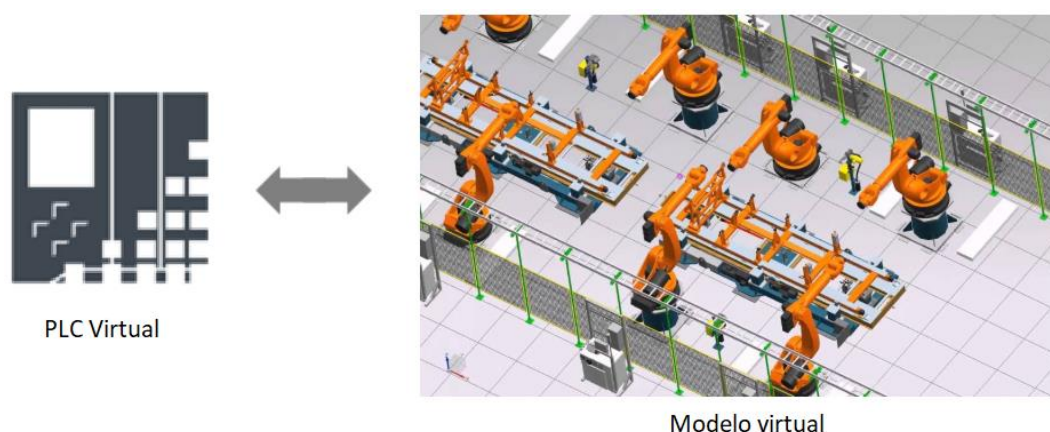


Figura 9 – Simulação *Software-in-the-loop* de uma linha de produção

O *HIL* representa a simulação com *hardware* embutido, isto quer dizer que, é utilizado um controlador físico, que pode ser um *PLC* real, onde é testado o verdadeiro processador do sistema, juntamente com o modelo virtual do processo físico (ver Figura 10). Este tipo de simulação, pode ser usado durante a execução deste modelo e, consoante os parâmetros definidos nos elementos constituintes do *HIL*, pode se modificar a maneira de trabalhar do sistema para atingir o seu correto funcionamento. Com o *HIL* pode-se verificar no mundo virtual os movimentos mecânicos dos componentes, os sinais elétricos, as electroválvulas, entre outros, para que seja efetuada a programação do controlador, visualizar e implementar no processo real [46].

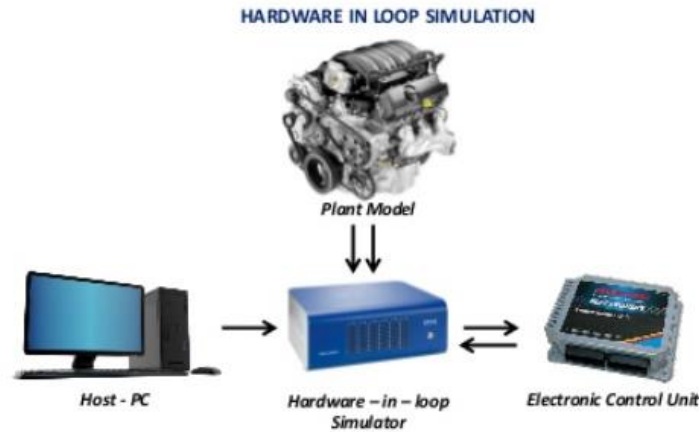


Figura 10 - Simulação *Hardware-in-the-loop* de um motor e o seu controlo elétrico

● Exemplos de *Virtual Commissioning*

A técnica do *SIL*, tem vindo a ser aplicada na área automóvel, no sentido de existir uma plataforma digital para criar modelos digitais de vários veículos e explicar a estrutura utilizada para a criação desses veículos consoante os ambientes a que estes vão estar sujeitos ao longo da sua utilidade. Isto ajuda os utilizadores a validar e comparar os diferentes tipos de operações dos modelos que pretendem fabricar [47].

Para mostrar a potencialidade do VC através do *SIL*, foi utilizado o *software Tecnomatix Plant Simulation* que simula um ambiente 3D onde é possível verificar, os movimentos e o design do processo elaborado, em conjunto com o *TIA Portal*, correspondente à programação, e o *PLCSIM* que simula um *PLC* virtual. O processo era apenas caracterizado pelo *start/stop* de um tapete rolante, mas o objetivo principal era de demonstrar as potencialidades deste *software* para realizar o *SIL* [48].

Outra aplicação utilizada do *SIL*, foi no desenvolvimento de uma tecnologia de satélite e o baixo custo para o elaborar, denominada por *Move-II CubeSat*. O *SIL* aplicado neste processo foi de simular o algoritmo de controlo da leitura de sensores, implementado em código de programação *C++*. Nesta aplicação é utilizado o *Simulink*, que é um *software* que permite a programação por blocos em sistemas de controlo e recorrer a vários tipos de simulações. Nesta aplicação também foi utilizado o *HIL*, no sentido de usar o *hardware* do satélite, com o objetivo dos algoritmos serem aplicados no *hardware* e não na simulação virtual. O *HIL* permite que o satélite interaja com um ambiente espacial simulado, pois existem modelos virtuais que substituem os sensores e atuadores [49].

O *HIL* também foi usado num projeto que tinha como objetivo testar um *software* de controlo industrial, através da conexão de um *PLC* a um simulador de eventos, denominado por *FlexSim*. Este simulador permite definir sensores e ações semelhantes ao comportamento do sistema físico, que neste caso é uma linha de produção industrial (ver Figura 11). Foi desenvolvido um *OPC Server* e uma plataforma *JAVA*, para estabelecer interligação entre o *PLC* e o *FlexSim*. Este processo demonstra a possibilidade de utilizar um *PLC* real (*hardware*) interligado ao *software FlexSim*, para simular um modelo virtual de um processo físico, que interage e responde aos sinais trocados através desta comunicação [46].

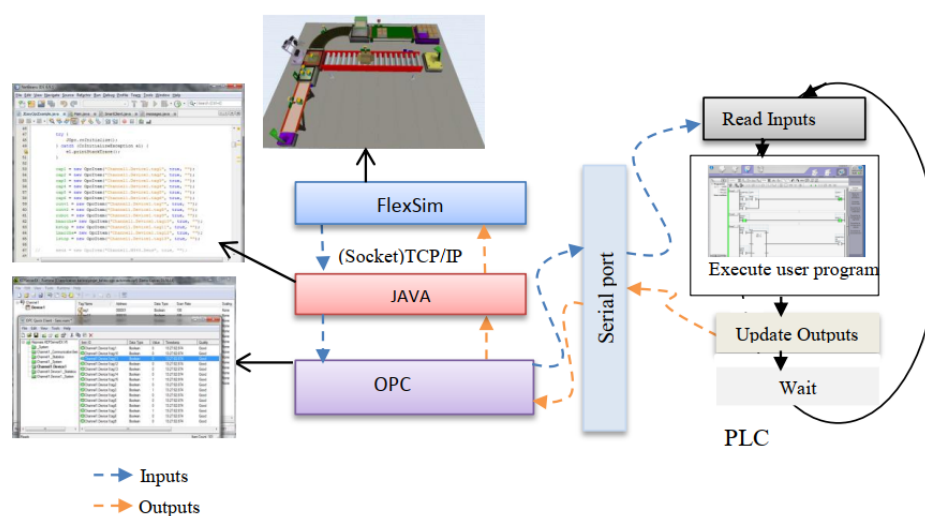


Figura 11 – Interface de comunicação na simulação *HIL* de uma linha de produção industrial [46]

2.5 Software de modelação e programação

Com o avançar dos anos, novo *software* e equipamentos foram desenvolvidos para serem aplicados na área da indústria, tendo em conta as práticas novas que iam surgindo. É o caso do aparecimento de *PLCs*, na altura da 3ª revolução industrial, que foram criados com o objetivo de substituir componentes elétricos como os relés e os contactores. Os *PLCs* permitem uma maior flexibilidade, para tornar os processos mais dinâmicos, consoante o tipo de função a desempenhar, o que com relés tornava tudo mais complexo. O *PLC* é constituído por um processador e módulos de entradas e saídas, e pode executar tarefas de controlo lógico, temporização, contagens, operações matemáticas, etc. Na programação deste equipamento, atualmente, utiliza-se *software* específico, que permite aplicar várias linguagens de programação. Marcas como *Omron*®, *Siemens*® e *Allen-Bradley*® são utilizadas em *PLCs* aplicados na indústria, e cada uma tem o seu modo de comunicação e de programação, embora com linguagens *standard* (norma IEC 61131-3) [50][51].

Na elaboração de projetos industriais, relativamente a sistemas físicos, surgiu a necessidade de realizar o desenho e modelação 3D de processos relativos a moldes, linhas de produção e montagem, máquinas, robôs, protótipos e outros. Engenheiros mecânicos e desenhadores, necessitavam deste tipo de recurso, o que levou a que fosse desenvolvido *software* para este tipo de função. Este *software* é constituído por operações, que permitem a aquisição de vários tipos de geometria, cor, textura, e outros, sendo possível atingir o objetivo de modelação 3D de um processo [52].

Seguidamente, descreve-se o *software*, para virtualização de processos e programação de equipamentos associados, dando destaque ao *software* que foi, efetivamente, utilizado neste trabalho.

2.5.1 Programação PLC

O *software* utilizado na programação de PLCs possibilita variadas tarefas de lógica de programação, tendo cada marca o seu modo diferente de programação e execução. Cada *software* permite definir distintos protocolos de comunicação industrial, para controlar dispositivos conectados em rede como HMIs, variadores eletrónicos de velocidade, SCADAs entre outros. A marca Siemens®, é provavelmente a que tem o maior número de dispositivos na indústria, a nível mundial, no que toca a PLC's [51].

Sendo o projeto em parceria com a Siemens Automation Academy, existe *software* e *hardware* da marca Siemens®, que se encontra disponível nesta academia para utilização no projeto. O *software* atual para programar, não só, autómatos da Siemens®, mas também outros dispositivos da mesma marca, é o TIA Portal. Este *software* permite executar várias tarefas de automação, no qual vai ser o responsável pela lógica de programação, do sistema descrito do projeto, e da interligação com o modelo virtual. É um *software* com potencial na área de automação que permite configurar redes de comunicação e controlo de supervisão [53]. O TIA Portal pode ser utilizado em autómatos reais, como no caso deste projeto, onde foram utilizados a gama S7-1200 e S7-1500, ou o PLCSIM, *software* que permite criar autómatos virtuais.

2.5.2 Criação de modelos virtuais

Para efetuar o modelo digital de um processo físico, é essencial determinar todas as medições e dimensões necessárias, para entender o funcionamento dos componentes físicos e obter uma melhor implementação no processo digital. Sensores, atuadores e outros elementos importantes, devem ser bem implementados no modelo para que funcionem da mesma maneira que na realidade [54].

Existe diverso *software* para efetuar a modelação 3D e a simulação de modelos virtuais em determinados processos. Para além do desenho, é necessário também a simulação 3D, que permite visualizar o comportamento de um processo físico com sensores e atuadores [54]. É utilizado *software* como o *Siemens NX*, que permite efetuar várias operações para modelar peças de uma determinada geometria física, definir juntas e restrições, tipos de materiais, etc [55].

Associado a este *software*, está o *Mechatronics Concept Designer (MCD)*, onde é possível visualizar o funcionamento do modelo digital a nível mecânico, elétrico e de fluídos para verificar o desempenho deste. O *MCD* tem um vasto conjunto de métodos para a implementação de como deve funcionar um sistema automático, como operações, sensores e propriedades físicas dos componentes. Tem incorporada a tecnologia *NVIDIA Physx*, onde é possível a simulação da física em tempo real, com todos os componentes que possam exercer forças, colisões, juntas de movimento, entre outros, tornando a simulação o mais realista possível. Pode se verificar a simulação de todos os componentes modulados, definindo o tipo de operações a executar, estando conectado, por exemplo, a um *PLC* [56].

Este *software* possibilita a interligação com autómatos da *Siemens*[®], através do *TIA Portal*, por via de vários protocolos de comunicação. O *Siemens NX* é o *software* utilizado para a realização do modelo virtual deste projeto, pois pertence à marca *Siemens*[®], e está disponível na academia para utilização.

2.5.3 Exemplos de *software* de simulação

Na realização de um *DT*, é necessário efetuar um modelo virtual através de um *software* 3D, que permita o desenho da sua forma física e o teste de movimentos físicos, referentes a esse processo.

O *Factory I/O*, é um *software* criado pela empresa *Real Games*[®], onde se realiza a implementação gráfica 3D de unidades fabris, colocando todos os componentes, em conjunto com atuadores e sensores. Este *software* é limitado nos componentes que este contém, pois não é possível criar um objeto com uma específica configuração, sendo necessário adaptar os elementos pré-existentes do *software*, de maneira a ser o mais parecido com o processo real. O objetivo da empresa *Real Games*[®] é de criar um *software*, que permite elaborar um processo automático industrial, através da conexão com o *software TIA Portal* e outros semelhantes, auxiliando no conhecimento na área de automação, nomeadamente na programação de *PLCs* [57].

Na Figura 12, está apresentada uma linha de produção com transporte de paletes, existente numa empresa. Foi elaborado um processo, com uma interligação entre um autómato *S7-1200*, programado no *TIA Portal*, com o *Factory I/O*, onde é constituído por vários transportadores [58].



Figura 12 – Linha de produção com *Factory I/O* [58]

Outro *software* com potencial para implementação de *DT*, é o *RobotStudio*, que pertence à marca *ABB*[®]. Este *software* é caracterizado por um ambiente de simulação gráfica, onde é possível implementar robôs da marca *ABB*[®], e definir: trajetórias consoante as juntas do robô, programação com linguagem *RAPID* e comunicação com outros dispositivos, através de vários protocolos. Permite importar desenhos em 3D, para serem usados no ambiente de simulação, podendo interagir com o robô, e assim, simular um processo físico real ou a instalar no futuro. Embora no *RobotStudio* seja possível integrar modelos 3D de objetos criados externamente com uma determinada configuração e geometria, este apenas é dedicado a robôs da marca *ABB*[®] [59].

Na Figura 13 encontra-se um projeto desenvolvido na ESTG com o *RobotStudio*. É um sistema automático que replica virtualmente o sistema automático real, existente no laboratório de

robótica da ESTG. No ambiente do *RobotStudio*, foi criado o modelo virtual deste sistema, adaptando os robôs para a marca *ABB*[®], e criou-se vários objetos, como os transportadores, para simular os seus movimentos e interagir com os robôs [60].

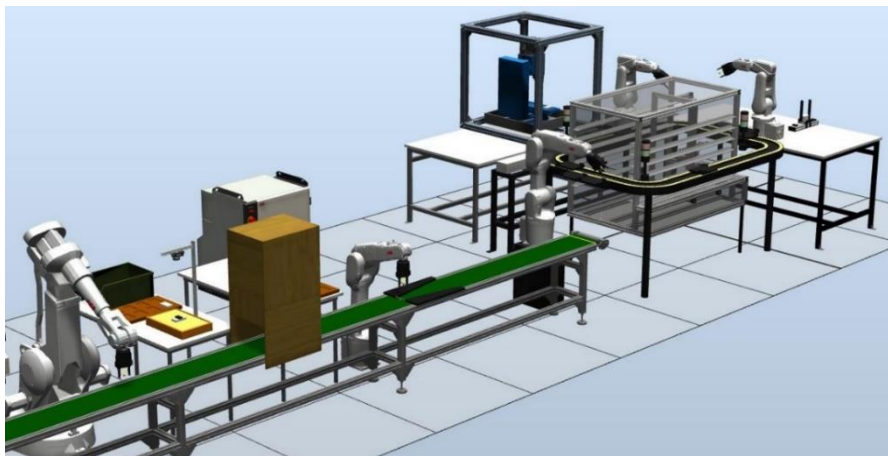


Figura 13 – Linha de montagem com *RobotStudio* [60]

O *software Emulate 3D*, que pertence à marca *Rockwell Automation*[®], é uma ferramenta dedicada a simular, de forma digital, sistemas de automação industrial. Foi desenvolvida para engenheiros na área da indústria o utilizarem, no sentido de ser aplicado em projetos de desenvolvimento industrial nos setores de armazém, produção, distribuição e transporte [61].

O objetivo do *Emulate 3D* é de criar um *DT* dinâmico, poupando tempo e dinheiro na realização de testes de controlo dos sistemas, antes de serem implementados. Este *software* é idêntico ao *Factory I/O*, no sentido de existir objetos já criados, neste caso, num catálogo, que podem ser utilizados num ambiente industrial, como transportadores e robôs. Este programa pode ser interligado a sistemas de controlo externos, normalmente *PLCs*, através de protocolos de comunicação, que permitem a aquisição de dados, como o *TCP/IP*, por exemplo. Na Figura 14, está demonstrado o ambiente virtual deste *software* [61][62].

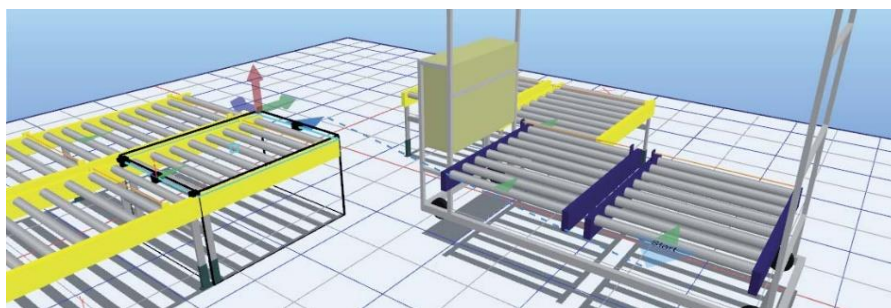


Figura 14 – Sistema industrial com *Emulate3D* [62]

Um *software* desenvolvido pela *Dassault Systèmes*[®], denominado por *Delmia*, oferece a capacidade de promover o mundo virtual da modelação e simulação, para fornecer uma solução completa na área da indústria. O *Delmia* tem a possibilidade de elaborar modelos 3D, incluindo ferramentas de transportes como *AGVs* e transportadores. Na simulação, é possível

utilizar a linguagem de programação *SCL* e *BCL*, onde se pode efetuar a lógica de programação das ações do modelo [63].

Este *software* foi utilizado num projeto para criar um método automático de trajetórias de transportes de AGVs. Foi desenvolvido um modelo 3D com um *Layout* específico e a sua programação, de maneira a obter automaticamente os caminhos de transporte a percorrer, do AGV (ver Figura 15) [64].

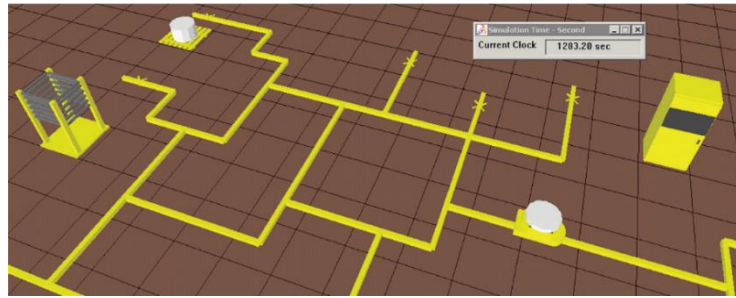


Figura 15 - Sistema automático com AGV no Delmia [64]

O *Siemens NX* permite elaborar o desenho e a simulação de um sistema industrial, como já foi explicado anteriormente, daí ser aplicado neste projeto. É possível realizar cada objeto individualmente, definindo o seu tipo de geometria e o seu papel no processo físico, para depois utilizar o *MCD*. O *MCD* executa a simulação de movimentos, onde se agrupa todas as partes, obtendo o modelo digital do processo físico [55][56].

Na Figura 16, é demonstrado um exemplo de projeto, onde a programação deste modelo foi elaborada pelo *TIA Portal* e foi utilizado um autómato da gama *S7-1500*. Implementou-se o protocolo de comunicação *OPC Classic*, onde o *PLC* cria um *OPC Server* e o *MCD* vai ser o *OPC Client*. O projeto em descrição, é constituído por um tapete rolante, sensores indutivos, botões, e indicadores luminosos e sonoros para eventuais eventos de alarme e de funcionamento do processo [44].

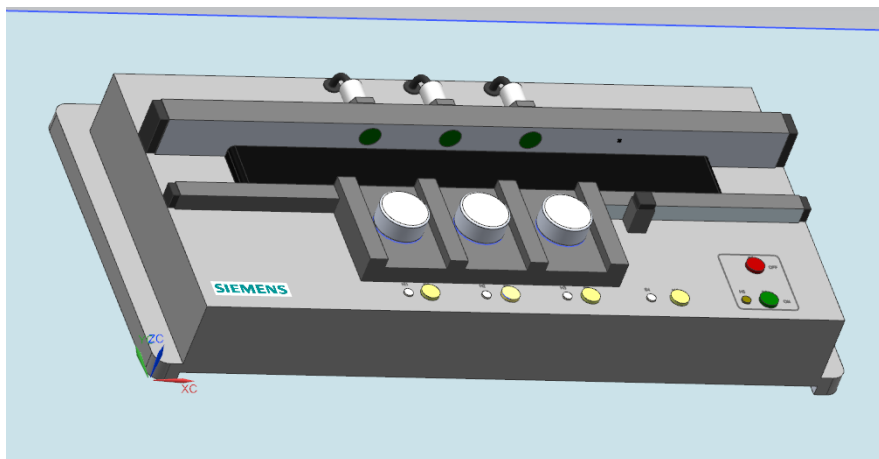


Figura 16 - Projeto de Digital Twin no Siemens NX [44]

2.6 Métodos de interligação e simulação

No desenvolvimento de um modelo 3D, deve-se utilizar ferramentas de *software*, que permitam executar a simulação de física, para verificar a precisão do modelo. Este método, é importante para apurar aspetos como colisões, forças, encaixes, movimentos e folgas. Se existir *software* de modelação 3D, com simulação integrada, é possível testar parâmetros nos componentes, para verificar o seu funcionamento. Por sua vez, em situações mais complexas, pode ser necessário interligar outro *software*, neste caso de programação, que possibilite a lógica de funcionalidade desse modelo digital. Num caso complexo, o *software* de programação permite um maior controlo sobre o funcionamento do modelo digital e ajuda a perceber a forma em como todo o processo irá funcionar, retificando alguns aspetos através da análise visual da simulação [65].

No que toca a simulações com interligação de outro *software*, podem ser usadas plataformas virtuais que permitam o controlo dos processos (*SIL*), ou *hardware*, como *PLCs*, utilizando protocolos de comunicação (*HIL*). As plataformas virtuais não necessitam de recorrer a qualquer tipo de *hardware*, e assim, não se corre o risco de danificar qualquer tipo de material, como é o caso do *PLCSIM*. Este cria um autómato virtual da *Siemens*[®], podendo ser aplicado para a validação da lógica de programação, do sistema implementado, e esse mesmo programa ser usado no mundo real. Possibilita a prevenção das tarefas automáticas do sistema e, aquando a instalação, utilizar a mesma topologia de programação. Apenas será necessário alterar alguns pormenores do sistema, no terreno, devido à capacidade de processamento do *PLCSIM* e do *PLC* real serem diferentes, e também do modelo virtual poder ter ligeiras diferenças com o mundo real [66].

Na utilização de *hardware*, pode ser utilizado um *PLC* real, que controla a programação de um modelo virtual, para perceber, a nível de processamento, o seu verdadeiro resultado. O programa utilizado na programação do modelo virtual, pode servir para aplicar no dispositivo físico correspondente a esse modelo, desde que o modelo respeite exatamente a componente real. Isto pode fazer poupar tempo aos técnicos de automação ou até informar clientes, de forma visual, como irá funcionar o processo mais concretamente [67][68].

Existe a possibilidade de utilizar também, o mesmo controlador (*PLC*), para o modelo virtual e para elementos do processo físico, por forma a serem executados em simultâneo, obtendo vários objetivos. Pode ser relevante definir especificações no modelo virtual, que tenham impacto direto no mundo físico, e vice-versa. Através da verificação do sincronismo dos dois

mundos e criando um sistema de supervisão com valores importantes, é possível que estes contribuam no desempenho do sistema. Neste caso, podem ser utilizados protocolos de comunicação, como *TCP/IP*, *Profinet*, *OPC*, etc. que permitem a troca de dados entre o mundo real e o digital. Uma rede de comunicação deve ser bem configurada, pois podem existir problemas no fluxo de dados entre o *PC*, que simula o modelo virtual, e um *PLC*, causando problemas na simulação [43] [44].

2.6.1 Protocolos de comunicação

Variadas marcas de *PLCs*, utilizam protocolos de comunicação diferentes, para interligar estes a dispositivos externos, ou como este projeto, a modelos virtuais 3D para simulação. Com o aparecimento do protocolo *Ethernet*, as marcas de produção de autômatos, decidiram utilizar esta como base nas suas comunicações, desde a *Profinet*, *TCP/IP*, *Ethercat* ou a *Ethernet/IP*. A *Ethernet* tornou-se numa rede de comunicação industrial dominante, pois é uma solução com boa performance e rápida na transmissão de dados, permitindo interligar vários protocolos *Ethernet* na mesma rede de comunicação, ao contrário da comunicação série (*RS-232*) [69].

Neste projeto em concreto, o equipamento aplicado é da marca *Siemens*[®], em que os autômatos têm como protocolos predominantes a *Profinet* e a *Profibus*. No entanto, o conceito da interoperabilidade específica a comunicação entre dispositivos de várias marcas que utilizam diferentes meios de comunicação. É neste contexto que entra o *OPC*, definido por um protocolo de comunicação *standard*, que especifica a troca de informação numa rede industrial, entre outros protocolos de comunicação de vários fornecedores, [70].

Sendo que cada fornecedor de *PLCs* tem o seu próprio protocolo de comunicação, não é simples a interligação entre todos os sistemas numa unidade fabril, daí surgir a necessidade de criar um protocolo *standard* como o *OPC* (*Open Platform Communication*). Nos anos 90, empresas na área de automação e a *Microsoft*, reuniram-se para formar a *OPC Foundation*. Como resultado dessa reunião, criaram a norma *OPC* (*OLE - Object Linking and Embedding for Process Control*), que permite o acesso a dados em tempo real entre os processos com o sistema operacional *Windows* e dispositivos físicos implementados nas fábricas. O primeiro padrão criado foi o *OPC Classic* que só acedia a dados, sendo depois desenvolvida para acumular informação e identificação de eventos, que correspondem a três especificações [71]:

- *OPC DA (Data Access)* – Troca de dados entre dispositivos como valores analógicos, tempos e contagens, que contribuem para análise de qualidade dos processos;
- *OPC HDA (Historical Data Access)* – Acumulação de dados para obter um histórico onde é possível consultar e analisar esses;
- *OPC A&E (Alarms and Events)* – Troca de informações de alarmes, eventos e variáveis discretas para efetuar a gestão de tarefas.

O *OPC* é um protocolo do tipo *Client Server*, que significa que existem um ou mais servidores numa rede, para fornecer dados. Estes dados encontram-se estruturados num modelo de informação, e aguardam que diversos *Clients* executem a interação com esse modelo fazendo pedidos ao *Server* para a obtenção dos dados. O *OPC Server* faz a conversão do protocolo de comunicação utilizado pelo *PLC*, para o protocolo *OPC*, disponibilizando informação. O *OPC Client* pode ser um *HMI* ou *SCADA*, que se conecta com o *Server*, para solicitar a troca de informação entre dispositivos [72][73].

Com o avançar do tempo, este protocolo teve uma boa aceitação por parte dos fornecedores de processos industriais, sendo utilizado em diversos dispositivos. Evita usar equipamentos físicos como *gateways* ou *bridges*, que seriam necessários para interligar os diferentes protocolos de comunicação. No entanto, surgiu o problema de ser restringido apenas a componentes que tenham incorporado o sistema operacional *Windows*. Assim, em 2006, foi criado o *OPC UA* que veio trazer mais vantagens em relação aos anteriores como o facto de não ser dedicado apenas ao *Windows*, ter segurança integrada, modelos de informação mais organizados e a possibilidade de encontrar servidores e clientes na mesma rede [70] [71][72].

2.6.2 Exemplos de utilização do protocolo *OPC UA*

O protocolo *OPC UA* tem vindo a ser utilizado e avaliado em unidades fabris, serviços e projetos de investigação, para o controlo e transmissão de dados de certos processos que necessitam da troca de informação entre vários setores, tendo como objetivo alcançar o conceito da Indústria 4.0.

Na Figura 17, encontra-se um exemplo de aplicação do protocolo *OPC UA*, numa linha de montagem com um robô. Esta linha é composta por uma máquina *CNC*, que é utilizada para fins educativos, um robô da *ABB*[®] e o controlador da linha de montagem. Para elaborar a rede de comunicação baseada no *OPC UA*, foi utilizado o *software C++ UA Server SDK* e o *.NET UA Client* obtidos pela *Unified Automation*. A comunicação entre o *OPC UA Server*, criado no controlador da linha de montagem, e o controlo numérico da máquina, é com base nos protocolos *DNC* e *TCP/IP*. O *OPC UA Client* vai ser utilizado no *software .NET UA Client*, conseguindo aceder a informação do *Server* e elaborar tarefas da máquina [74].

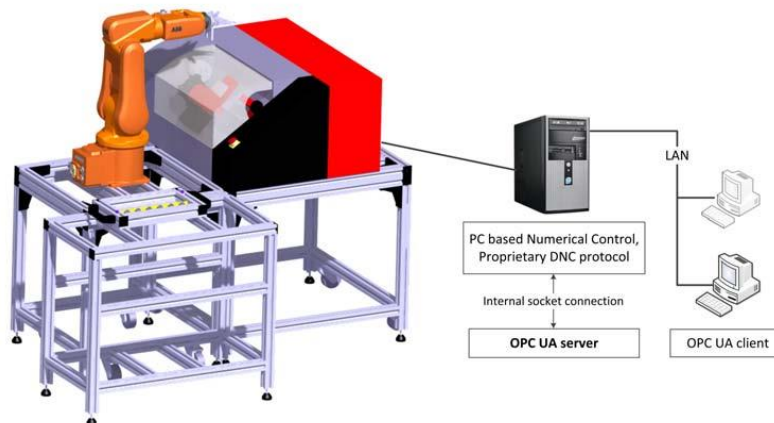


Figura 17 – Linha de montagem robótica com *OPC UA* [74]

Outro projeto também desenvolvido no âmbito do protocolo *OPC UA*, teve como objetivo a realização de uma plataforma *web*, que representa um sistema *HMI*. É usado o protocolo *OPC UA* para efetuar a comunicação entre o *PLC* e o *AJAX*, uma ferramenta que permite criar páginas *web* interativas. Na Figura 18, encontra-se o esquema de comunicação desenvolvido, que consiste num *OPC Client*, permitindo que a plataforma *web* aceda ao *OPC Server*. Este *Server* contém informação do controlador do processo a ser monitorizado. Nos resultados obtidos, os autores deste projeto de exemplo, verificaram que a troca de informação na plataforma é um pouco lenta, logo é indicada apenas para sistemas que não necessitem de tempos de supervisão instantâneos. No entanto, pode facilitar na monitorização de processos, a nível remoto [75].

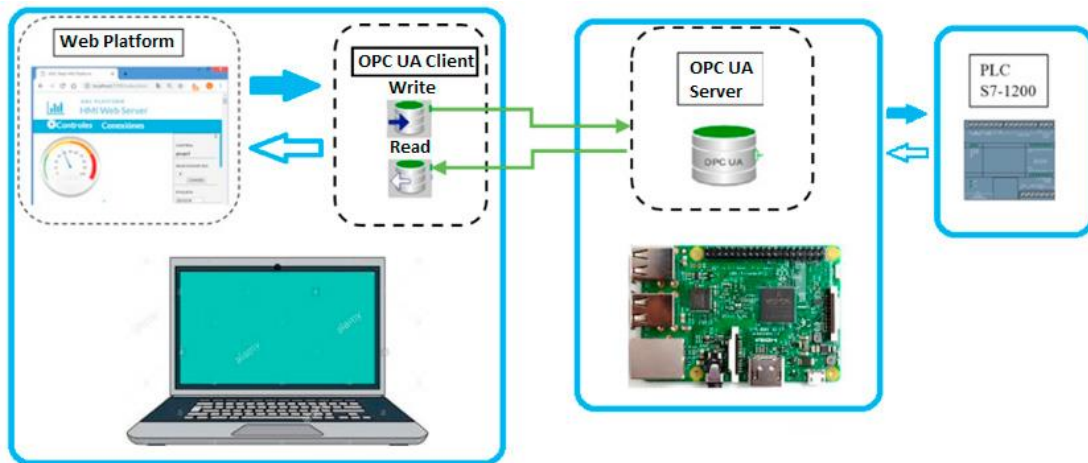


Figura 18 – Plataforma web com OPC UA [75]

No projeto ilustrado pela Figura 19, foi implementado um controle de um processo de laboratório, que simula um industrial, com a utilização de tecnologias *AutomationML* (*Markup Language*) e *OPC UA*. O *AutomationML* permite a transmissão de dados de forma *standard*, em sistemas de produção industriais, com base em formato *XML*. O processo laboratorial é composto principalmente, por um robô e um tapete rolante, que comunicam entre si. O robô recebe o seu controle através de um módulo *Wi-Fi* e uma comunicação série, e o tapete rolante é controlado por um *PLC*. Um computador com o sistema operacional *Windows* foi utilizado para criar o *OPC UA Client* usando o software *JAVA-based OPC UA client* [76].

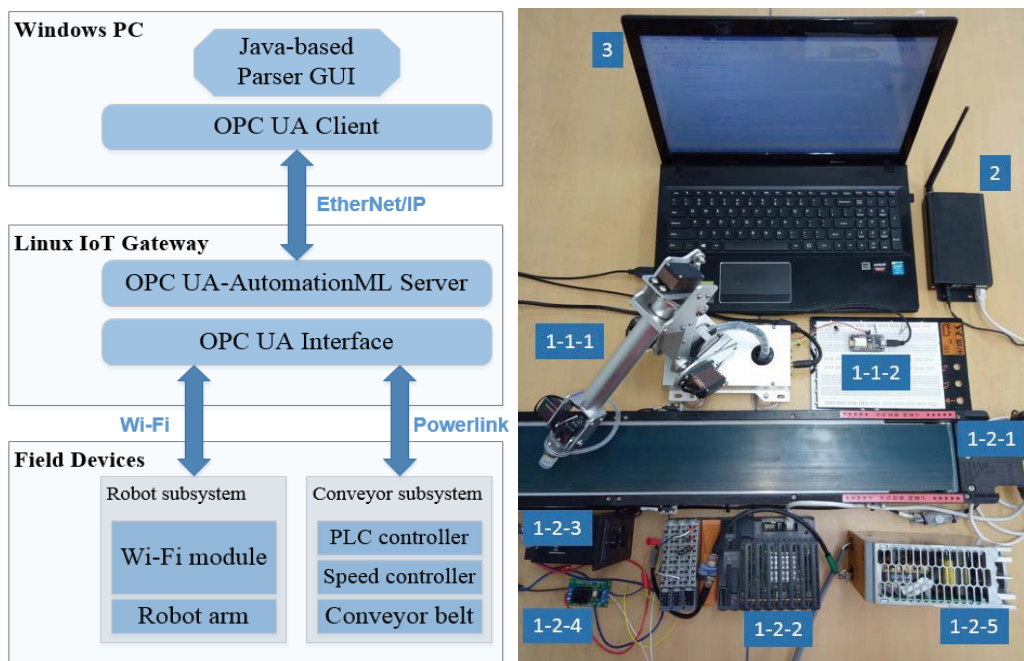


Figura 19 – Processo laboratorial com Automation ML e OPC UA [76]

Foi utilizado o *OPC UA* para o desenvolvimento de um controlo de monitorização, num sistema de refrigeração. Foi aplicado o *software MATLAB®* que permite criar, não só um sistema de supervisão, mas também implementar algoritmos. O estudo engloba a comunicação entre um *PLC*, o sistema de refrigeração e o *MATLAB®*. Conseguiu-se atingir uma boa comunicação, obtendo a qualidade de sinal em tempo real. Na Figura 20, encontra-se o sistema de monitorização aplicado neste projeto [77].

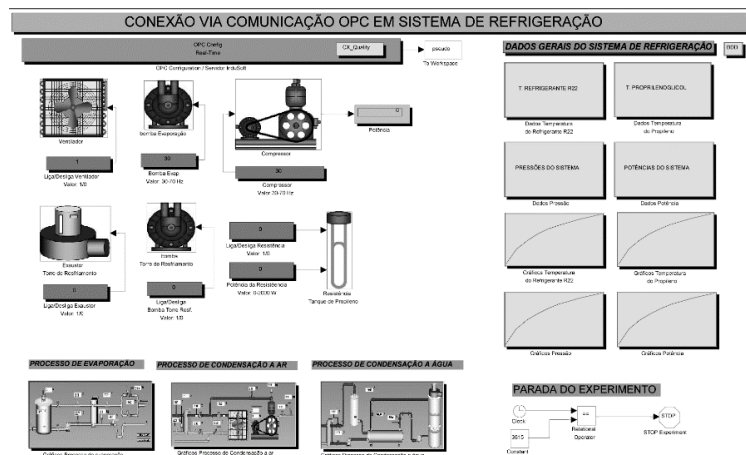


Figura 20 – Sistema de monitorização com *MATLAB®* e *OPC UA* [77]

Para além do protocolo *OPC UA* ser usado na indústria, também pode ser usado noutras áreas. Um projeto na área da agricultura foi elaborado, no âmbito de adaptar este protocolo industrial, para um controlo e monitorização remota de uma máquina agrícola, aplicando o conceito do *IoT*. A máquina agrícola usada neste projeto, já tem incorporada uma rede *CAN* como os automóveis atualmente, e um sistema de monitorização com *GPS* e comunicação *RS-232*. Na Figura 21 (esquerda), verifica-se que foi elaborada uma interface com o *software LABVIEW* para interligar dados do sistema de monitorização e a rede *CAN*. Um *OPC UA Client* foi criado também, para verificar o sinal de *GPS* do veículo remotamente (Figura 21 lado direito) [78].

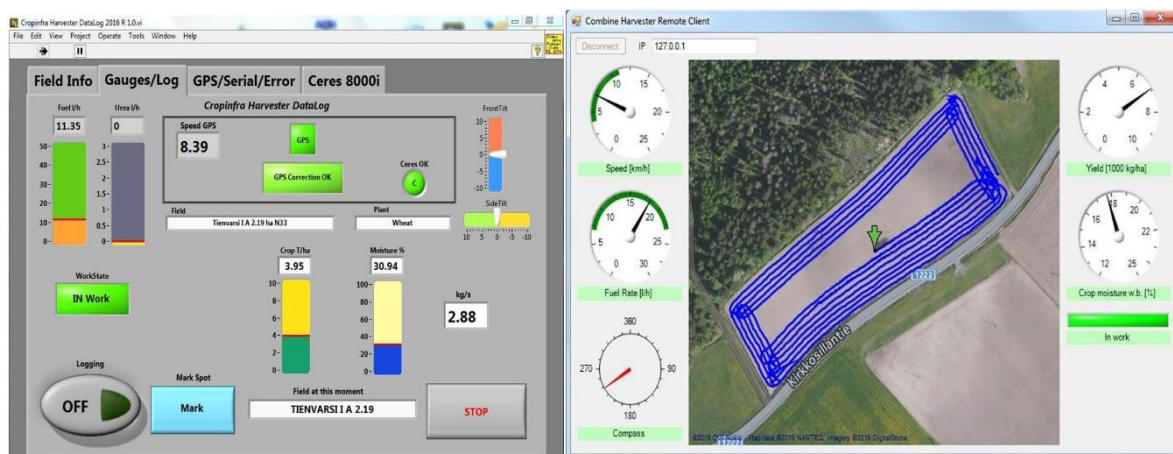


Figura 21 – *Labview* e *OPC UA Client* numa máquina agrícola [78]

3. Desenvolvimento

Como já foi referido, o objetivo deste trabalho, é o de realizar um *DT* de um *kit* existente na *Siemens Automation Academy*, e efetuar a sua programação e simulação. A realização deste modelo virtual pode auxiliar, neste caso, para o monitorizar, verificando o seu correto funcionamento e efetuar o controlo. Com este procedimento, existe a possibilidade de uma comparação com o processo real, no sentido de melhorar o seu funcionamento.

O desenvolvimento deste projeto teve várias fases como:

- Perceber o funcionamento do processo real;
- Elaboração do modelo virtual, consoante as características do *software Siemens NX*;
- Avaliar a potencialidade de conexão entre o *software Siemens NX* e *PLCs*;
- Elaboração de testes para atingir o propósito deste projeto.

Vai ser abordada a descrição técnica realizada, bem como os problemas e resultados obtidos, tendo em conta o tema do trabalho que é a virtualização de processos na Indústria 4.0. Todo o *software* e *hardware* utilizados neste projeto, estão disponíveis na academia da *Siemens*[®], na ESTG.

Através de pesquisas na Internet, fóruns e artigos, foi possível encontrar processos idênticos, que utilizam os mesmos tipos de técnicas e permite auxiliar na elaboração deste projeto, obtendo um conhecimento mais completo. Ao longo do desenvolvimento deste projeto, novas ideias surgiram, e novos objetivos seriam definidos, em conjunto com os orientadores. Assim, foi possível exercer um maior estudo sobre o *DT*, utilizando as ferramentas de *software* e *hardware* deste projeto.

3.1 Processo físico

Inicialmente, o projeto começa pela identificação do funcionamento do processo físico, sendo neste caso uma mesa de montagem rotativa, utilizada nas aulas de automação industrial. Assim sendo, será importante perceber como todos os seus componentes físicos trabalham a nível mecânico e elétrico. Ao efetuar este procedimento, é possível auxiliar na definição dos objetivos do projeto e verificar algum tipo de anomalia que possa existir, para a tentar melhorar, com a elaboração do modelo digital e a sua interligação com o processo real.

3.1.1 Descrição do processo físico

O *kit* é uma mesa de montagem de peças criado pela empresa *Bytronic*. É constituído por uma mesa rotativa, que fornece um conjunto de peças já montadas a um operador, onde este transfere manualmente para um tapete rolante. O tapete rolante não faz parte deste *kit*, sendo que será apenas a mesa rotativa a ter em conta. Na Figura 22, verifica-se o processo físico em questão, onde se pode observar que é constituído por seis postos de montagem, e as peças a armazenar nos postos, são pretas ou brancas, podendo conter duas peças cada posto [79].

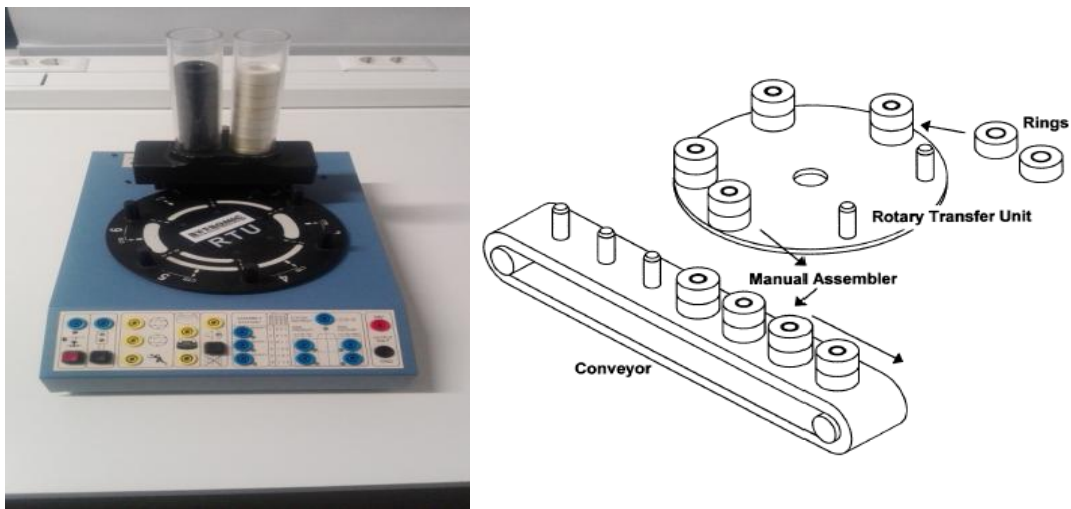


Figura 22 – Mesa de montagem rotativa

A nível de sensores, este *kit* contém acoplados sensores óticos e infravermelhos, que auxiliam na aquisição de dados. É o caso na identificação de cada posto, visto que são seis postos, existem três sensores de posição infravermelhos (**p2**, **p1** e **p0**), em que, através da combinação binária destes, é possível atribuir qual o posto associado, como se vê na Tabela 1. Para além disso, contém um outro sensor (**c**), para indicar que o posto está centrado, no local onde são colocadas as peças, recebendo-as de forma adequada.

Posto de montagem	p2	p1	p0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

Tabela 1 – Codificação dos sensores de posição

Quando a mesa se encontra centrada num posto, pronta para receber peças, é possível colocar duas peças em cada posto, existindo dois sensores **a1** e **a2**, que indicam se a peça está na posição inferior ou superior, respetivamente. Para distinguir a cor da peça, são utilizados sensores óticos, **c1** e **c2**, que permitem identificar se a peça que foi recebida no posto é branca, mas caso o sensor não for ativo, significa que a peça é preta.

No caso dos atuadores deste processo físico, existe um motor associado à mesa rotativa que permite que esta se movimente no sentido horário (**Mh**) ou no sentido anti-horário (**Mah**), e ainda outro sinal que dá a possibilidade de colocar uma velocidade mais rápida (**F**) na mesa. Na colocação de peças nos postos de montagem, existe um comutador que se desloca para a esquerda ou direita, com retorno ao centro automático, fazendo cair por gravidade, as peças brancas (**Sb**) ou as peças pretas (**Sp**).

Para além dos sensores e atuadores, este *kit* também possui um painel que contém botões de comando, sinalizações e conectores. Os botões de comando são a ordem de arranque, emergência e inibição da sirene (**S**), que pode ser utilizada para quando existe um alarme. As sinalizações são respetivas aos sensores e os conectores servem para ligar os sinais de entradas (sensores) e saídas (atuadores), de um *PLC*. Na Figura 23, encontra-se descrito as funcionalidades deste painel de comando.

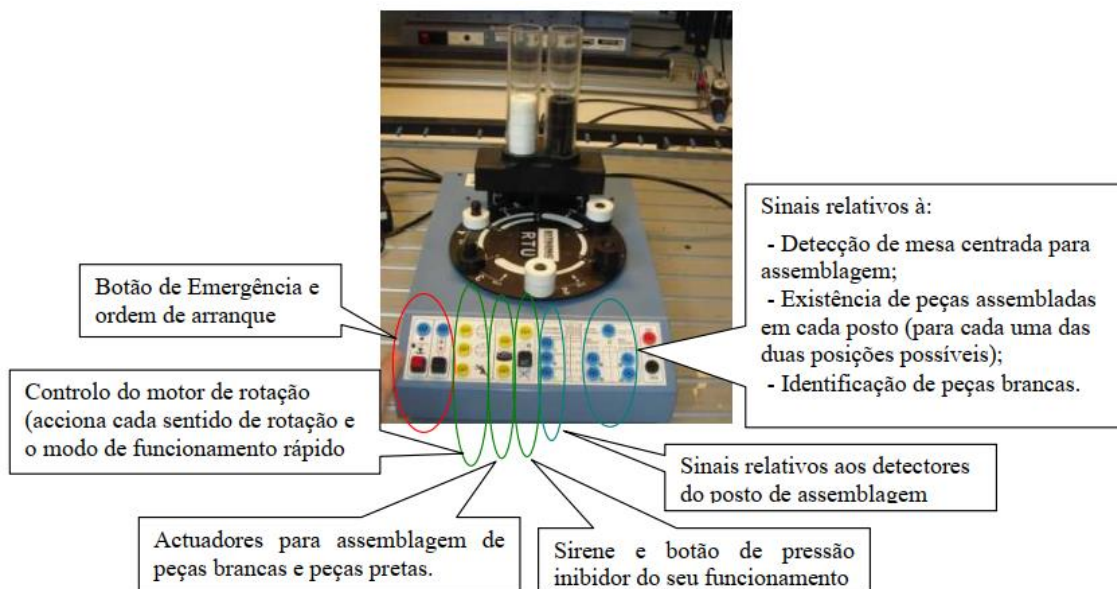


Figura 23 – Descrição do painel de comandos da mesa de montagem

Tendo em conta os componentes deste processo físico, iniciou-se a observação visual do funcionamento deste, onde não foi necessário qualquer tipo de programação de autómato. O procedimento para visualizar o funcionamento do *kit* foi de forçar eletricamente os atuadores, e verificar o comportamento dos sensores, os movimentos mecânicos, tempos de processo e outros aspetos importantes. Estas observações, iriam influenciar na criação do modelo digital, na programação e na sua interligação.

Em relação à parte de automatização deste processo, não existe um modo de funcionamento sequencial obrigatório, em que se possa identificar um objetivo. Em termos de testes e simulações, pode-se adotar uma determinada sequência para atingir um conceito de automatização.

No modo de funcionamento que se definir, deve-se ter em conta o facto deste processo ter de ser autónomo e capaz de definir certas otimizações de processo. Por exemplo, ao movimentar a roda para uma determinada posição, identificar qual o melhor sentido de rotação a executar, bem como a utilização da velocidade mais rápida ou não. Assim, é possível efetuar movimentos no menor tempo de processo possível, tendo em consideração que estes aspetos não afetam o bom funcionamento do processo, e de maneira a colocar as peças nas posições corretas.

3.2 Modelo virtual

De acordo com o tema deste projeto, era necessário a elaboração de um *DT*, através da criação de um modelo virtual do processo físico já descrito. Um *DT* é sempre associado a um processo físico, seja ele existente, ou não. Neste projeto o processo físico já está concebido, e nesse caso, o *DT* pode ter a capacidade de se interligar com um controlo de supervisão e monitorização digital. Esse *DT*, deve corresponder detalhadamente ao sistema físico, de maneira a desempenhar as tarefas que o processo físico executa.

O desenvolvimento do modelo virtual, foi das tarefas mais desafiantes deste projeto. A criação de um modelo virtual, exigia conhecimentos de mecânica e desenho 3D, facto pelo qual o autor não tinha qualquer experiência em *software* do género, tornando-se mais complexo, adquirir essa aprendizagem. Sendo assim, o *software Siemens NX*, foi explorado ao longo deste projeto para verificar como elaborar o desenho 3D do processo físico, e a empresa *CADFLOW*, ajudou nos conhecimentos bases deste *software*.

3.2.1 Siemens NX

Como já foi referido, para elaborar o modelo virtual de um processo físico, é necessário efetuar todas as medições necessárias de cada componente (ver Figura 24), e perceber o funcionamento deste. O modelo virtual elaborado foi uma cópia virtual do processo físico, onde o *software Siemens NX* permite criar, individualmente, os vários tipos de peças e depois agrupá-las num projeto só.

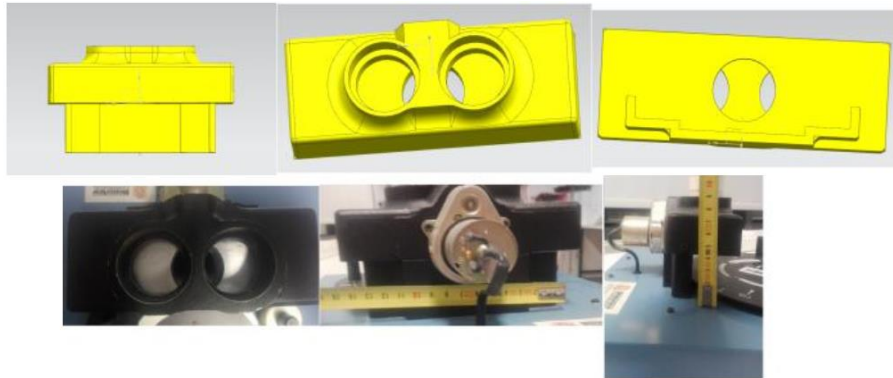


Figura 24 – Base do comutador de peças do modelo virtual e do processo físico

Em termos de componentes criados, este modelo é constituído principalmente, por uma base azul, uma roda com seis pinos, uma base onde são recolhidas e entregues as peças, um comutador de peças, as peças brancas e pretas e os tubos onde estas são colocadas manualmente. Na Figura 25, encontra-se o modelo digital, elaborado através do *software Siemens NX*.

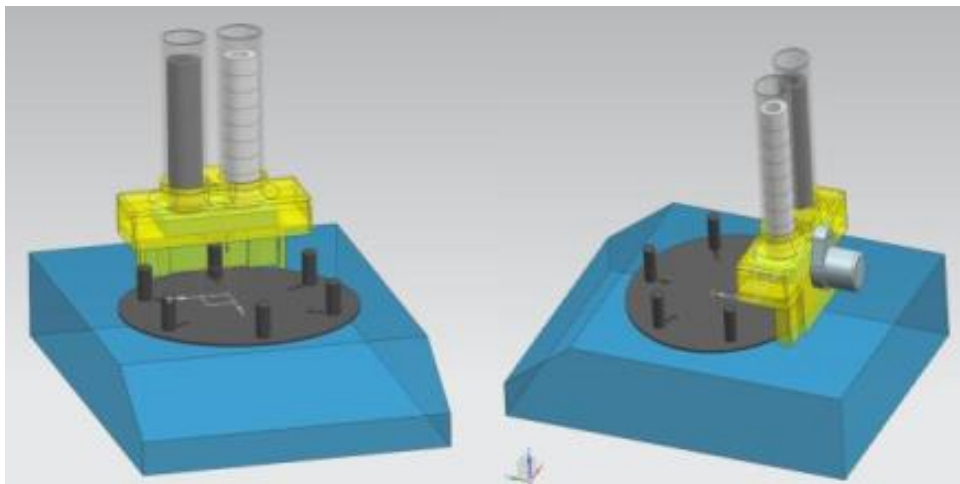


Figura 25 – Modelo digital da mesa de montagem rotativa

Pode se dizer que este modelo, replica o processo físico, exceto alguns pormenores que iriam contribuir no funcionamento da simulação, e vão ser demonstrados mais à frente, na parte da simulação. Por outro lado, existiram alguns detalhes no modelo 3D que foram melhorados, relativamente a anomalias encontradas no processo real, também descritas mais abaixo.

O ambiente do *Siemens NX*, permite criar peças através da técnica de modelação. Cada peça foi criada num modelo único, em que se pode elaborar um esboço da forma geométrica do componente pretendido como retângulos, círculos, linhas, perfis, etc. Com essa geometria criada, é possível efetuar operações como extrusão, furação, criação de bordas, chanfros, entre outros. Existem diversas formas e técnicas de elaborar este tipo de procedimentos, sendo que ao longo do projeto novas aptidões foram ganhas para melhorar os métodos implementados.

Na Figura 26, encontra-se um exemplo dos métodos elaborados para realizar o modelo de um componente da mesa de assemblagem. Foi necessário a realização de várias operações, que deram origem ao componente desejado. Este componente foi a primeira peça a ser desenhada, que é a base azul do processo físico.

Começando por seleccionar a opção para desenhar um esboço, foi efetuado um retângulo com as dimensões adequadas, onde posteriormente se efetuou a ação de extrusão. De seguida, aplicou-se um chanfro, que recorta uma zona entre duas faces. Devem ser seleccionadas as faces aplicadas, o tipo assimétrico, visto não ser igual para as duas faces, e as distâncias que se pretende cortar, de um lado e do outro. Por fim, aplicou-se uma furação, seleccionando o diâmetro desta e a profundidade.

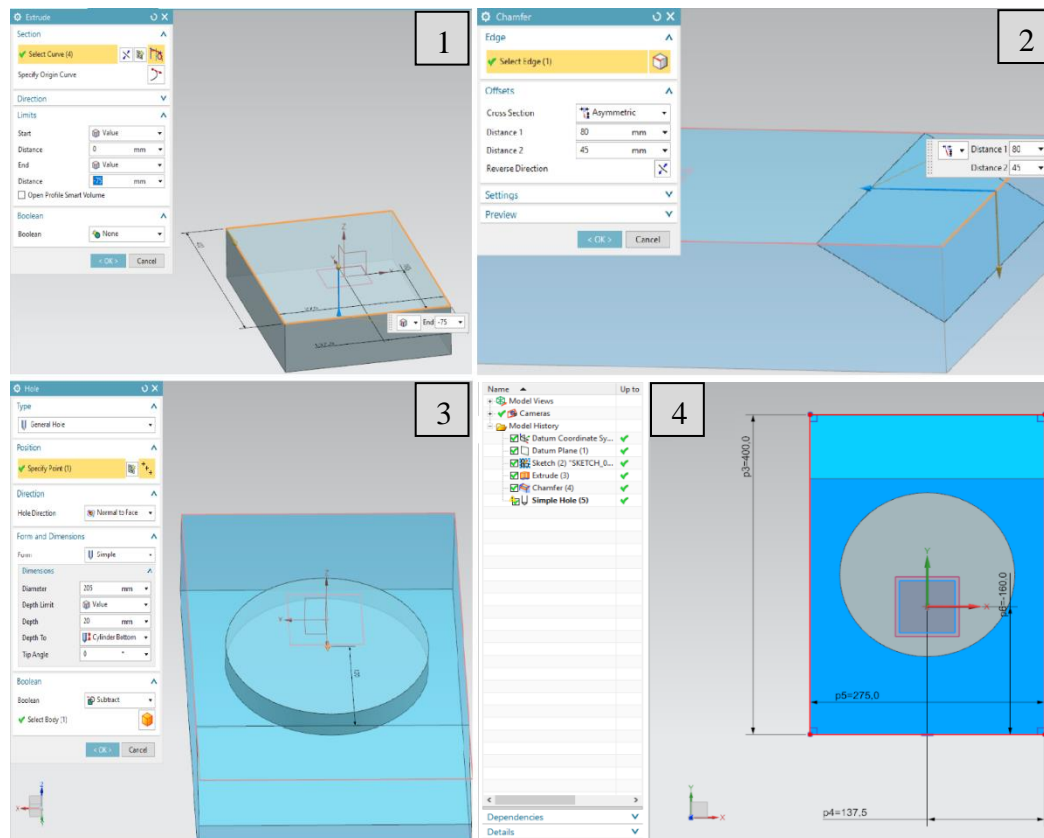


Figura 26 – Modelação da base da mesa de assemblagem

O *Siemens NX*, tem uma aplicação que é o *MCD*, para realizar a simulação de física do modelo virtual. O *MCD* permite efetuar uma grande variedade de ações, começando pelo facto de efetuar a montagem de todos os componentes, num modelo 3D de uma máquina, ou neste caso, da mesa de montagem rotativa.

É possível definir a questão do comportamento físico de cada componente, perante a simulação, principalmente a nível de corpos rígidos e de colisão. Para além disso, o *MCD* também tem um campo para criar juntas e restrições, onde se pode criar as juntas fixas, juntas articuladas ou juntas de deslizamento. O tipo de material associado a cada componente, é importante definir também, pois influencia na sua estrutura e, conseqüentemente, o comportamento na simulação vai ser diferente. Por fim, devem ser definidos os sensores e atuadores do processo, em que podem ser utilizados, na parte dos atuadores, o controlo de posição, velocidade, superfícies de transporte etc. No caso dos sensores, podem ser criados sensores de colisão, distância, posição, velocidade etc. Para testar o funcionamento do modelo definido, existe a secção de operações, onde é possível testar este, e aplicar melhorias ao longo da simulação, para ir ao encontro do funcionamento correto do processo físico.

Após a elaboração dos componentes do modelo virtual, procedeu-se à inicialização do *MCD*, onde foram adicionados os componentes um a um, na secção de montagem, colocando-os no local adequado, através da ferramenta de movimentação de componentes. É possível movimentar um objeto de forma dinâmica, onde se pode deslocar nos três eixos (X, Y, Z) ou rodar nos ângulos associados a esses. Também pode ser utilizada a distância ao longo de um eixo, ou até rodar um ângulo sobre um eixo entre outras opções. Na Figura 27, encontra-se a parte de montagem do modelo virtual.

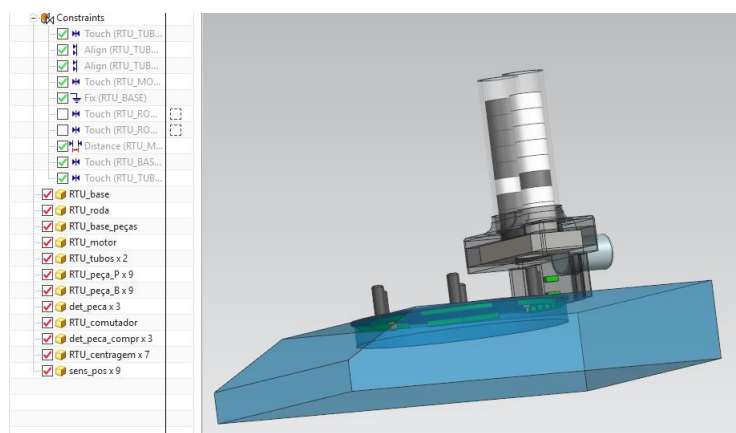


Figura 27 – Montagem do modelo virtual da mesa de montagem

Um fator importante na questão da montagem deste modelo, são as restrições ou condições mecânicas que os componentes têm uns com os outros. Essas condições podem ser partes fixas, alinhamentos, distâncias, toques entre dois objetos etc. Na Figura 28, encontra-se um exemplo com a base das peças, em que é necessário tocar na base principal.

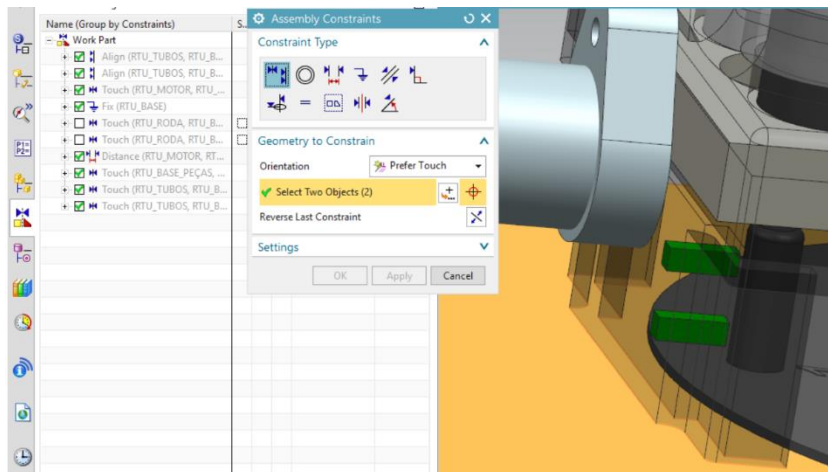


Figura 28 – Restrições mecânicas entre a base de peças e a base do modelo virtual

Estando definida a questão da montagem do modelo virtual, procedeu-se à secção da física dos objetos. Esta secção é das mais relevantes, no que toca ao funcionamento do modelo digital. Iniciou-se o procedimento com a física básica dos componentes, em que, um componente que tenha um comportamento na simulação perante a gravidade, tem de ser considerado um corpo rígido. Na criação de um corpo rígido, basta seleccionar o componente que é pretendido e é possível definir vários parâmetros físicos como o peso e a inércia, se for necessária maior precisão num modelo 3D. Neste caso, definiu-se o modo automático e pode se verificar na Figura 29, um exemplo de criação de um corpo rígido.

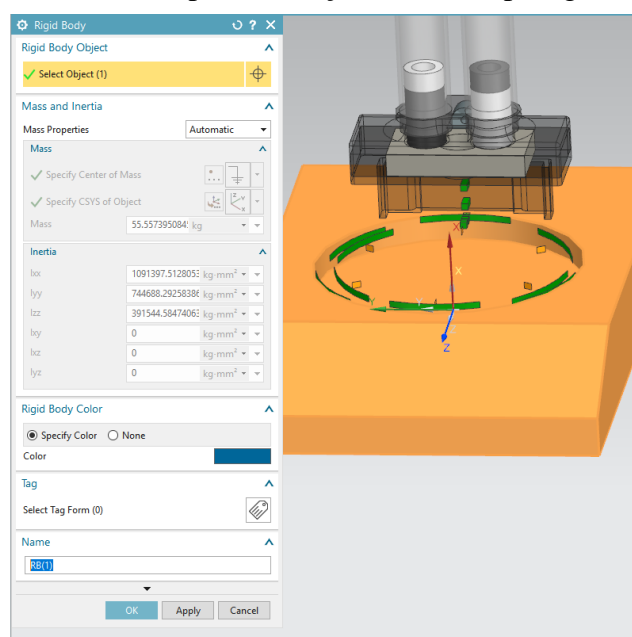


Figura 29 – Criação de um corpo rígido da base do modelo virtual

Ao definir um corpo rígido e realizar o arranque da simulação, este vai cair por gravidade. Para tal não acontecer, é necessário definir um corpo de colisão respetivo a esse corpo rígido, para colidir com outros objetos, ou criar uma junta fixa, que vai ser explicado na secção de juntas e restrições. Se for definido apenas um corpo de colisão, sem o corpo rígido associado, este não vai ter ação sobre a gravidade. Os corpos de colisão são mais sofisticados e requerem uma maior atenção na criação destes, pois contribuem uma grande parte, no desempenho da simulação. Ao criar um corpo de colisão sobre um componente, existem diversas formas geométricas que se podem seleccionar, como uma caixa, cilindro, esfera, convexo ou em malha. Também tem um fator convexo que melhora o detalhe do componente de colisão. Na Figura 30, encontra-se a criação de alguns tipos de corpos de colisão.

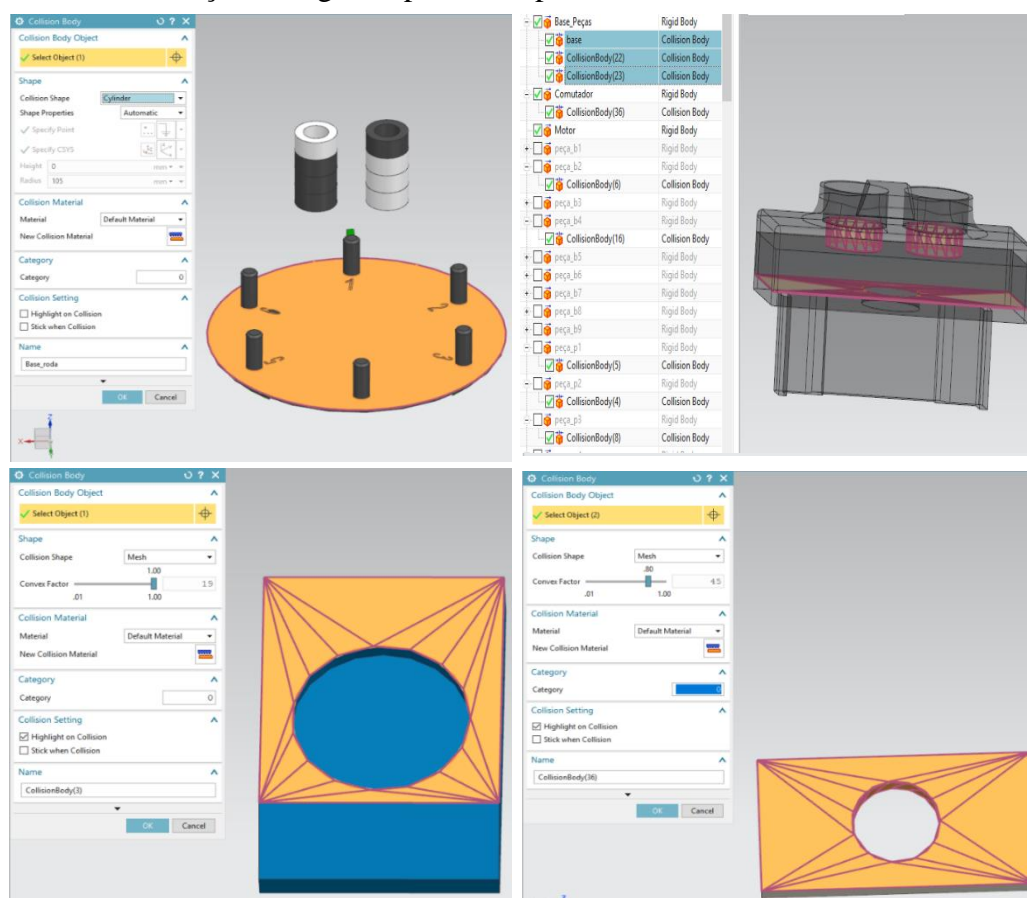


Figura 30 – Criação de corpos de colisão do modelo virtual

A próxima etapa a elaborar, é na secção de juntas e restrições. Começando por definir as juntas fixas, que é o mais simples, é possível criar um componente fixo, que se pode agregar a outro (base). A junta fixa define um componente fixo que não se move durante a simulação, como por exemplo a base azul. Um outro componente fixo pode estar também associado a uma determinada base, que se move juntamente com esta, como se pode verificar na Figura 31.

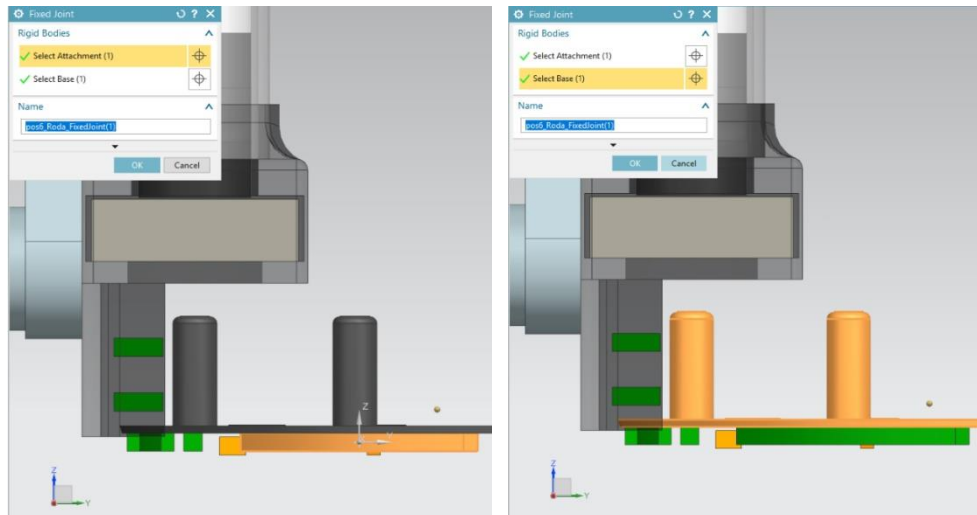


Figura 31 – Criação de junta fixa da codificação das posições com a roda como base

O comutador de peças, movimenta-se para esquerda ou direita fazendo cair as peças por gravidade. Neste caso, é necessário criar uma junta deslizante, que permite criar uma translação ao longo de um eixo. A Figura 32, apresenta os passos para realizar este tipo de junta. Verifica-se que é selecionado o comutador como componente principal, que tem como referência (base) a base azul, e movimenta-se ao longo de um eixo. Sendo assim, ao movimentar-se, o valor da sua posição ou velocidade é positiva para o sentido do vetor definido e negativa para o outro sentido.

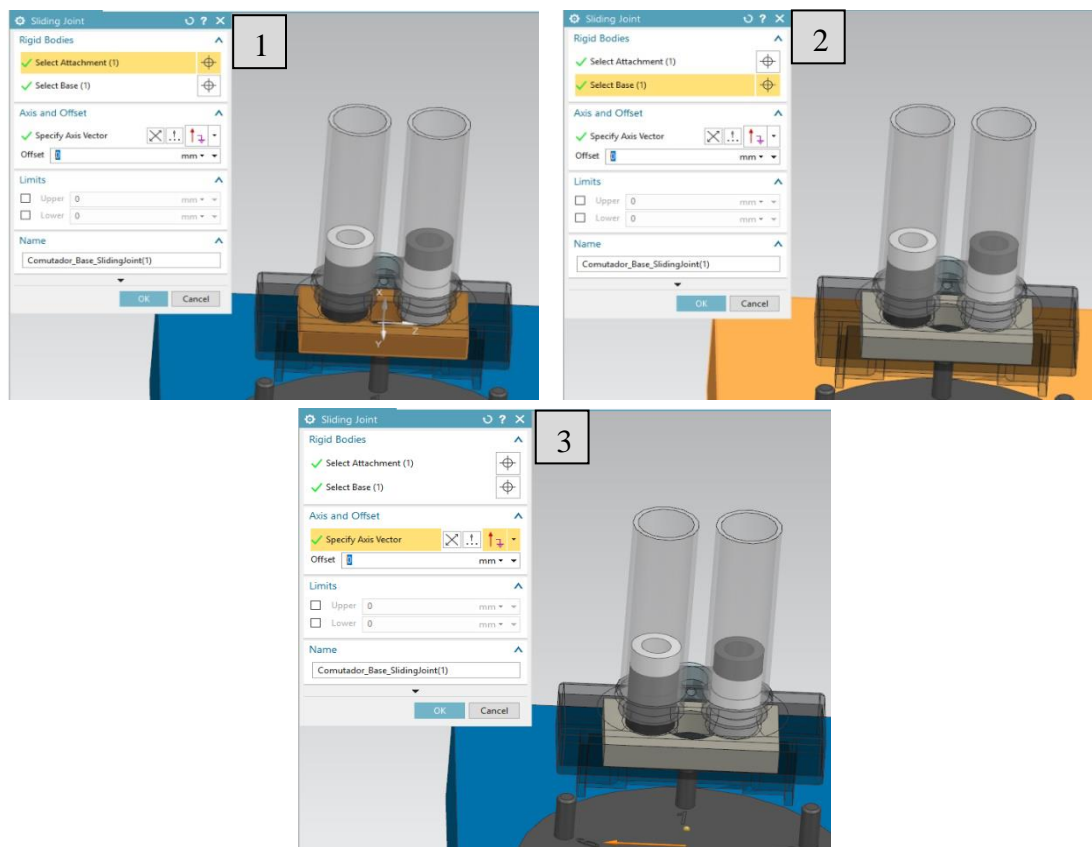


Figura 32 – Criação de junta deslizante do comutador de peças com a base azul como base

Outra junta criada, foi a junta articulada que gira sobre um eixo, neste caso a roda, que pode rodar no sentido horário ou anti-horário. A elaboração desta junta é parecida com a anterior, só tem o acréscimo de definir o ponto central no qual a roda deve rodar. A Figura 33 demonstra o sucedido.

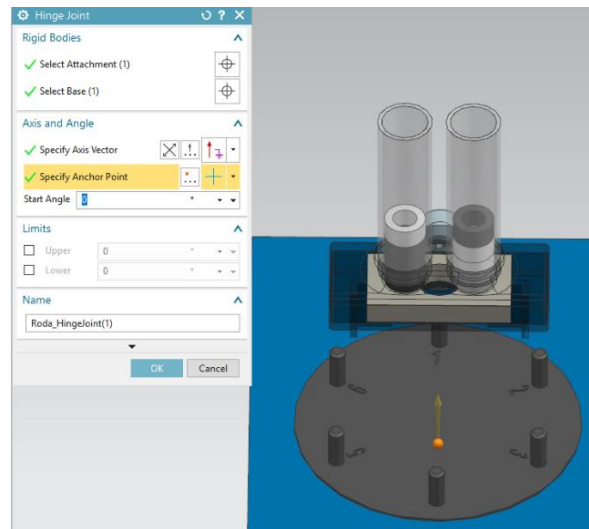


Figura 33 – Criação de junta articulada da roda

A próxima secção que se segue, é a dos sensores e atuadores do sistema. Os sensores só foram acrescentados mais à frente, na parte das operações, porque no começo, o fundamental era observar o funcionamento mecânico de todos os componentes, e iniciou-se pela definição dos atuadores. Tendo em conta as juntas criadas, foram definidos dois controlos de posição, um para a roda (graus) e outro para o comutador (mm). Este controlo permite definir na simulação, a posição para a qual se pretende que um componente se desloque. Na Figura 34, está a definição do controlo de posição para a junta deslizante associada ao comutador.

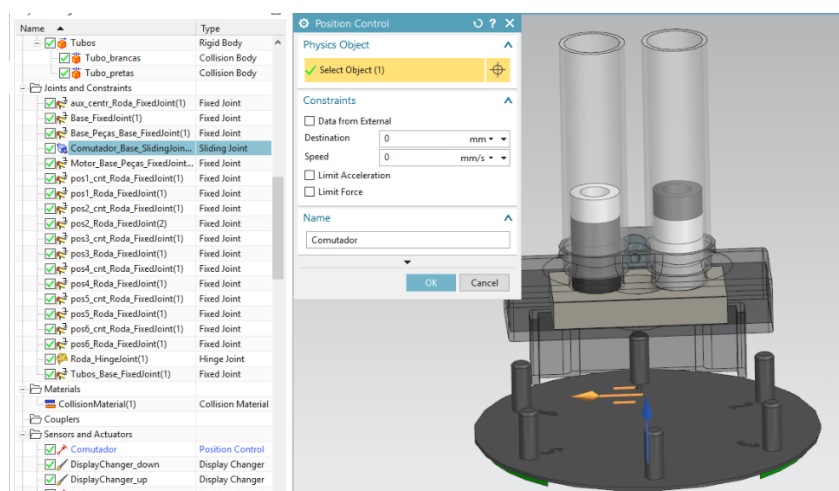


Figura 34 – Definição de controlo de posição do comutador

3.2.3 Operações no MCD

O MCD, para além das opções que contém já descritas, também permite a execução de operações. Isto permite verificar o funcionamento do modelo virtual, tornando-o mais preciso. Neste tópico, serão abordados os testes iniciais efetuados da simulação, os primeiros problemas observados e as alterações mecânicas que se efetuaram nos componentes, para corrigir esses.

Como já foi referido, este processo necessita de uma sequência de funcionamento, para atingir um determinado objetivo. Na Tabela 2 – Sequência de funcionamento, encontra-se a descrição do funcionamento atribuído ao DT.

Ordem sequencial	Posto de montagem	Tipo de peças
1º	3	2 brancas
2º	1	1º branca, 2º preta
3º	4	2 pretas
4º	5	1º preta, 2º branca

Tabela 2 – Sequência de funcionamento do DT

Consoante esta tabela, as operações podem ser iniciadas, tendo em conta os movimentos necessários para atingir o objetivo de funcionamento. Na Figura 35, está definida a primeira operação que é o controlo de posição da roda. A posição inicial é a nº1, e esta tem de rodar até à posição nº3. Sendo assim, visto serem 6 posições ao todo, a diferença entre cada posição é de 60°, logo é necessário rodar 120°. Também é possível definir a velocidade (°/s) da roda, ou definir o tempo que esta vai demorar a atingir 120°. Neste caso atribuiu-se um tempo, e seleccionou-se a velocidade automaticamente para atingir esse tempo definido.

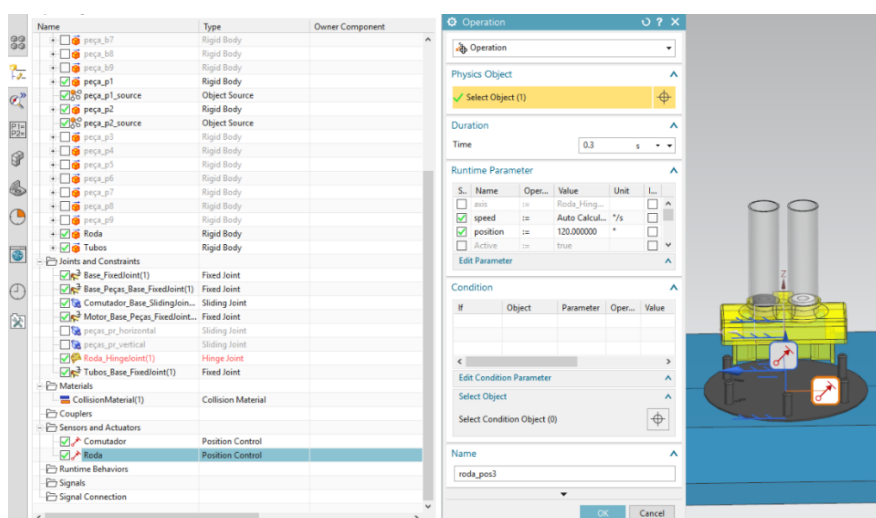


Figura 35 – Operação de posição da roda

Após atingir a posição nº3, é necessário colocar duas peças brancas no posto respetivo. Para tal, é necessário movimentar o comutador de peças, fazendo cair as peças por gravidade. O procedimento é semelhante ao anterior, só altera o tipo de grandeza física, em vez de ser em graus, é em milímetros, como demonstra a Figura 36.

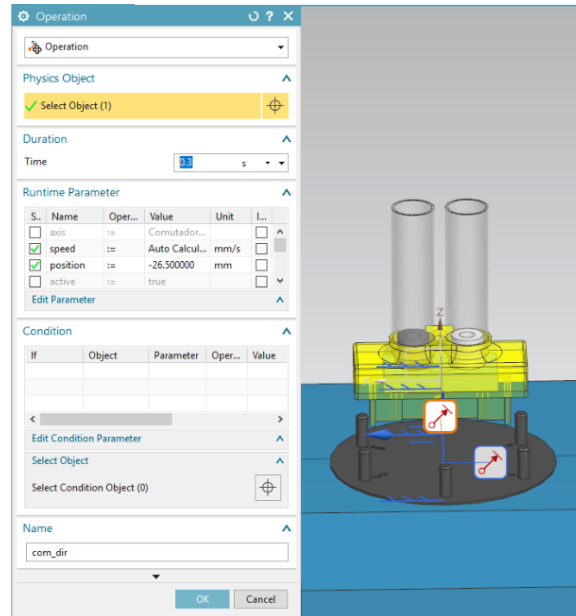


Figura 36 – Operação de posição do comutador

Considerando estes passos efetuados, o resto das operações foram realizadas e os testes iniciais observados, para verificar o correto funcionamento do modelo virtual. De seguida, descreve-se o início da simulação, os problemas encontrados, e as modificações realizadas para os resolver.

• Início da simulação

Ao iniciar a simulação, o primeiro lapso encontrado foi no acontecimento das peças serem expulsas dos tubos de armazenamento. É neste aspeto que é possível identificar a importância dos corpos de colisão. Inicialmente, os componentes que têm forma cilíndrica, que é o caso das peças, dos tubos e dos pinos da roda, foram definidos corpos de colisão com a mesma forma geométrica (cilíndrica). Mas tal não é possível, pois esse tipo de colisão, cria um cilindro fechado, e o que se pretende é que as peças sejam armazenadas no interior do tubo e colocadas nos pinos da roda. A melhoria que se aplicou, foi alterar o tipo de corpo de colisão para malha, sendo possível atingir esse objetivo descrito. Na Figura 37, é possível observar o sucedido. Também na base azul do modelo, o corpo de colisão não pode ser definido do tipo caixa, se não a roda fica presa a este e não executa movimentos.

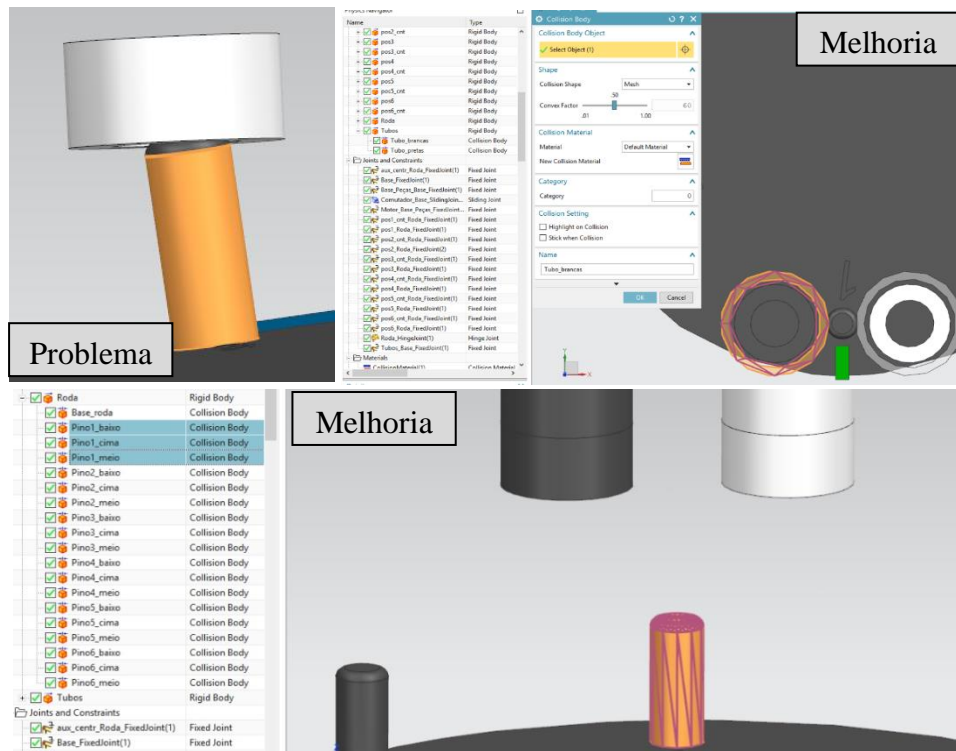


Figura 37 – Problema e melhoria nos corpos de colisão

Outra falha encontrada, foi no acontecimento de utilizar todas as peças na simulação. Ao todo são 18 peças, 9 de cada tipo, o que tornava a simulação do *MCD* demasiado lenta. Este fenómeno deve-se ao facto de algumas situações. Uma delas, é ao utilizar maior parte dos corpos de colisão do tipo malha. Este tipo de colisão permite melhor precisão geométrica, mas em termos de fiabilidade e desempenho de simulação é pior, o que requer maior poder computacional [80].

Também pode ser devido ao processamento do computador, usado na implementação deste *DT*, estar limitado. Este tem como características um processador *Intel(R) Core (TM) i5-6500 CPU @ 3.20GHz*, 8 GB RAM e sistema operativo *Windows 10 64-bit*. De acordo com pesquisas efetuadas, o que é aconselhado para obter melhor performance, com o *software NX MCD*, é utilizar um processador superior a este, o *Intel Xeon E5-1620 v3, 3.50 GHz* e memória RAM a partir de 16 GB [81]. Na Figura 38, pode se verificar que o *PC* está a exercer um grande esforço de processamento e memória, ao executar a simulação do *NX MCD* (assinalado a vermelho). Por vezes, exibe também um erro na memória interna, obrigando a reiniciar o *software*.

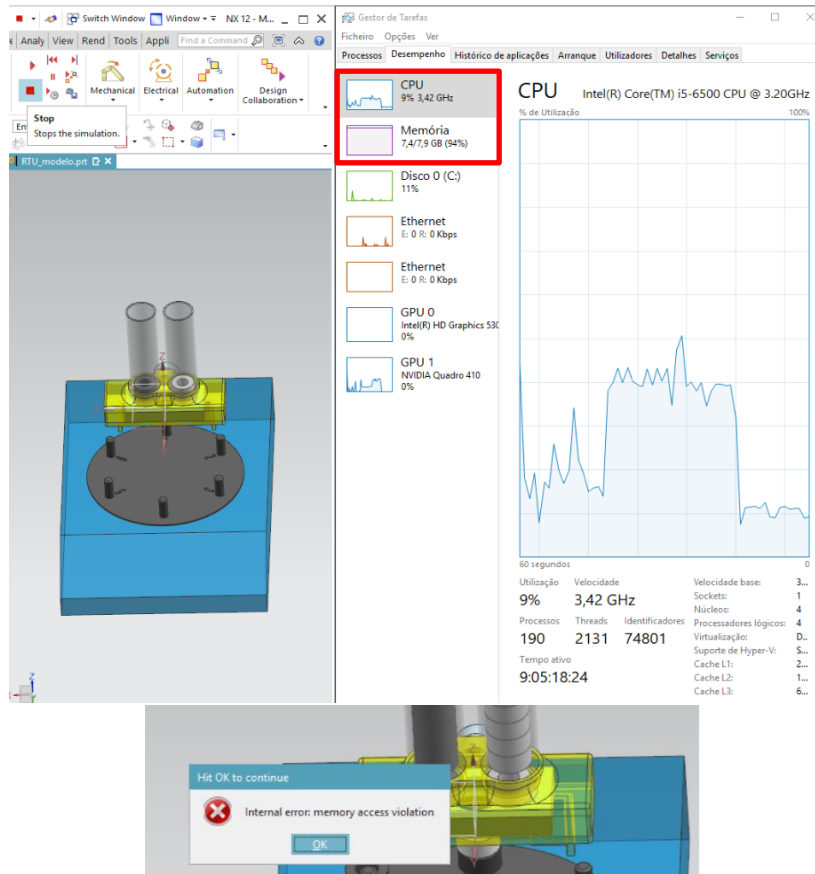


Figura 38 – Processamento do computador na simulação e erro NX MCD

Para além destas situações, o *NX MCD* também permite efetuar uma avaliação da performance de visualização. Esta ferramenta analisa o modelo construído, dando indicações onde é possível melhorar a performance da simulação, como se vê na Figura 39. Estas indicações serviram para proceder à seleção de opções, que efetuam essas melhorias indicadas, mas o problema ocorria na mesma.

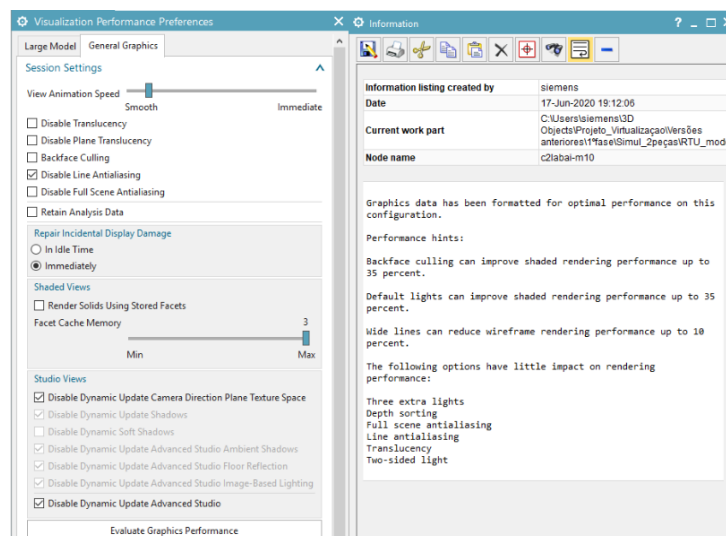


Figura 39 – Avaliação de performance dos gráficos no NX MCD

Devido a este problema encontrado, decidiu-se não considerar todas as peças na simulação e criar, na secção de física básica, uma fonte de objetos (ver Figura 40). Esta fonte gera objetos ao longo de um tempo, ou quando esta é ativa. Neste caso, apenas são ativas previamente, quando são necessárias colocar nos postos. Adicionalmente, uma operação teve de ser elaborada para ativar esta funcionalidade.

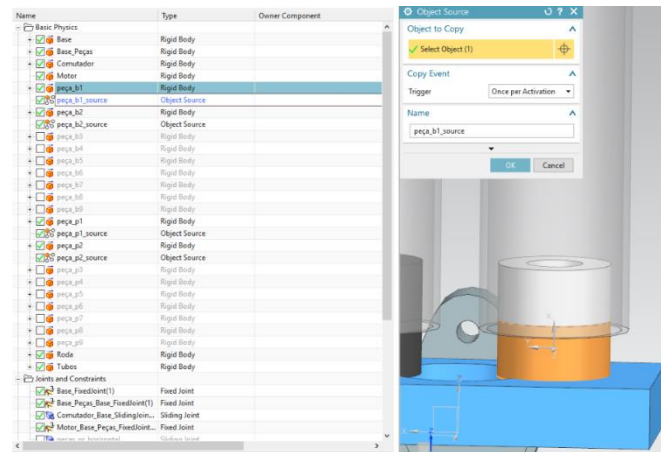


Figura 40 – Criação de fonte de objetos

Com estes dois problemas resolvidos, tornou-se possível executar uma parte da simulação, de forma razoável, no entanto ocorriam ligeiros conflitos na movimentação das peças. Estas não se comportavam de forma desejada, porque ao existir colisão, balançavam muito. Isto provocava serem expulsas da área de trabalho, ou não encaixavam bem nos pinos da mesa rotativa. Para otimizar esta movimentação das peças, optou-se por criar um novo material de colisão, associado a todas as peças, que determina o comportamento de colisão destas. Este novo material tem 4 propriedades, onde se define por critério e tentativa em erro, o valor adequado a cada uma. Essas propriedades são o atrito dinâmico, estático, rotativo e a restituição [82][83]. Tendo em conta estas considerações, definiram-se os valores do atrito estático e do rotativo, como se pode verificar na Figura 41, permitindo uma movimentação mais estável das peças.

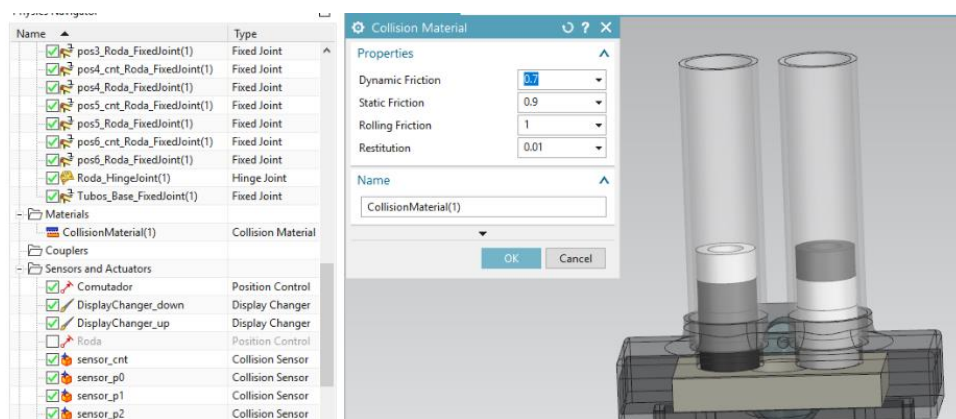
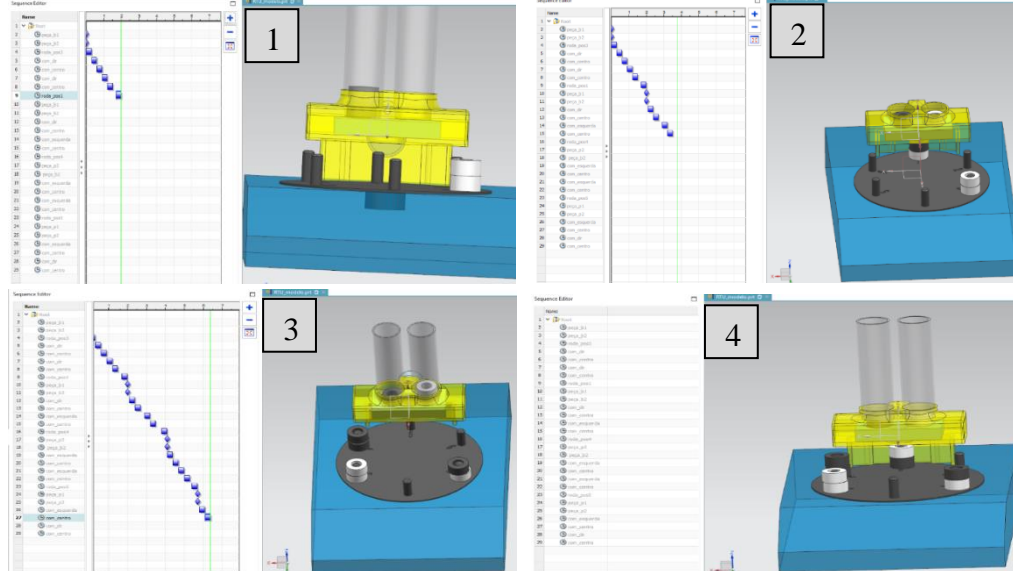


Figura 41 – Otimização das propriedades físicas das peças



onar corretamente, a nível de atuadores e

os sensores foram assim adicionados. Tendo em consideração que não existem sensores infravermelhos nem óticos no *software*, estes tiveram de ser adaptados no ambiente virtual. Como já foi referido, existem vários tipos de sensores que podem ser utilizados no *MCD*. Os sensores utilizados, para determinar a posição dos postos e de presença de peça inferior e superior num posto, foram adaptados para sensores de colisão. Ao longo da criação destes sensores, verificaram-se várias formas de os elaborar, de acordo com a funcionalidade destes. Os sensores óticos que permitem a distinção entre peça branca ou preta, foram criados em tarefas posteriores, devido a esta fase, ser de maior dedicação à simulação funcional, e não à deteção de anomalias, no sentido de identificar peças de cor errada.

Foi assim necessário criar os sensores, onde os que definem a posição dos postos, têm a codificação de 3 sensores, representados na Tabela 1 (tópico 3.1.1), o **p0**, **p1** e **p2**. Para além dos sensores criados, foram adicionados componentes no modelo, através do *Siemens NX*, que permitem a colisão com estes sensores e a codificação desejada. Adicionalmente, o sensor de centragem **c** e os outros sensores de deteção de peça inferior ou superior, também foram criados. Na Figura 43, está representada a tarefa de criação de um sensor de colisão e os componentes inseridos, assinalados a vermelho.

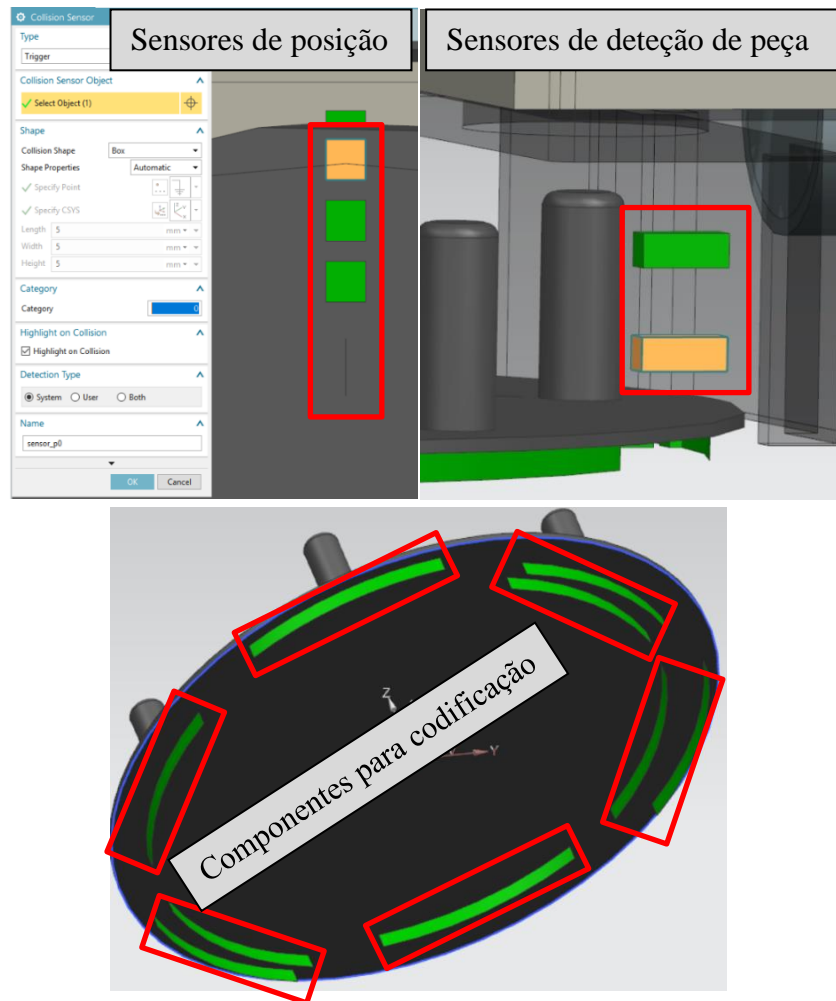


Figura 43 – Criação de sensores de colisão e de componentes adicionais

Com os sensores criados e testada a sua funcionalidade, procedeu-se às operações finais. Ou seja, em vez de serem baseadas em tempos, existe agora a possibilidade de existirem condições e eventos, que tornam o processo de simulação mais próximo da realidade. Sendo assim, foram realizadas operações, tendo em conta as condições destes sensores, para alcançar os postos de montagem corretos e para a deteção de peças nos postos. A operação, de fazer parar a roda na posição nº3, que é a inicial na sequência assumida, está indicada na Figura 44, e as condições indicadas com um retângulo a vermelho.

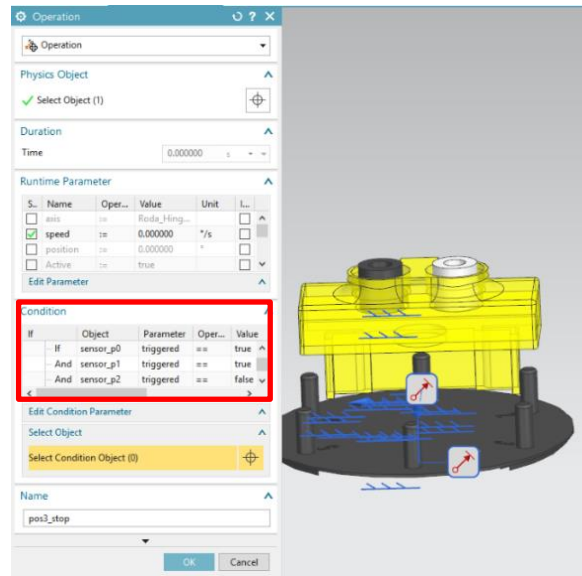


Figura 44 – Operação atingir posição 3 com condições de sensores

3.2.4 Criação de sinais e instruções

Estando a parte das operações testadas e confirmado o correto funcionamento do modelo virtual, seguiu-se a criação de sinais. Estes sinais criados no *MCD*, podem ser, posteriormente, atribuídos a componentes existentes no modelo, que permitem a interação com comunicações externas, dos vários meios de comunicação estudados. A forma como foram criados os sinais neste projeto, foi atribuindo o nome **IN_sinal**, para os sinais que o *MCD* recebe como entradas, de uma fonte de controlo externa, e **OUT_sinal**, para os sinais que o *MCD* envia como saídas, para essa fonte. As entradas podem ser, por exemplo, ordens para um atuador ser habilitado, o controlo de posição, velocidade etc. As saídas, podem ser os sensores, informação dos corpos rígidos, entre outros. Os sinais também podem ser de vários tipos de grandezas, desde booleanas, inteiras, ou reais, consoante a função desejada.

De seguida, apresenta-se na Figura 45, a criação de dois sinais booleanos. Um corresponde a uma saída, que envia informação quando o sensor **p0** é acionado, e outro, a uma entrada, para o comutador se deslocar na seleção de peças pretas (**Sp**). Pode-se verificar que na criação do sinal do sensor, é selecionada a conexão com um componente existente, sendo possível obter várias opções na função do sinal, para além do facto de ser acionado. Quanto ao sinal do comutador, este também poderia estar conectado com um componente, que por exemplo, seria o valor que iria controlar a posição deste, sendo necessário o tipo real para esse sinal. No entanto, decidiu-se efetuar uma instrução que controla essa posição, e daí ser necessário apenas criar um sinal booleano simples, sem qualquer interação direta com um componente existente.

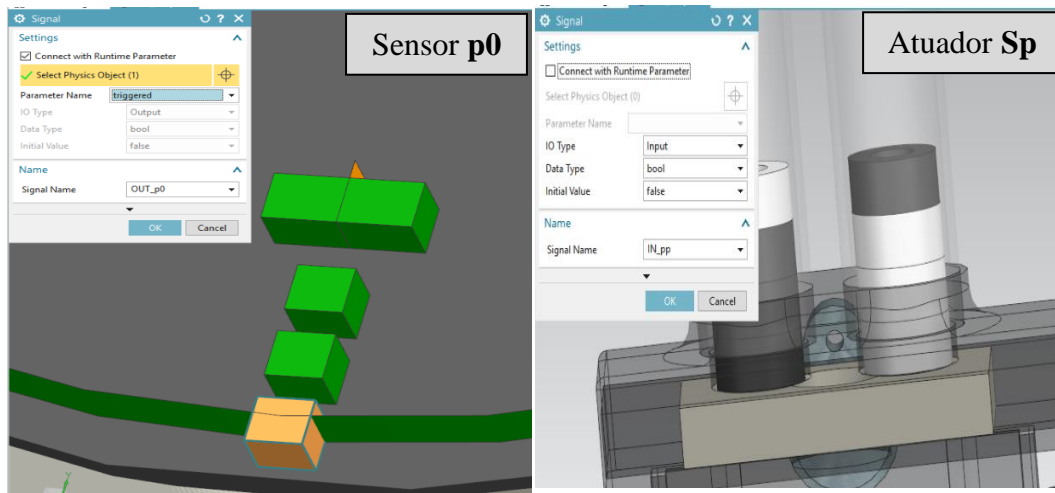


Figura 45 – Criação de sinais de posição e do comutador

Como já foi referido, o processo real tem dois sensores que possibilitam a deteção do tipo da peça, branca ou preta, para determinar a correta montagem desta. No entanto, o *software NX MCD*, não contém sensores óticos para efetuar essa função. Sendo assim, foi criado um sinal para cada peça, no sentido de existir essa distinção do tipo de peça montada. Cada sinal, está diretamente conectado com o componente, neste caso, o corpo rígido associado à peça, atribuindo o valor de centro de massa, no eixo Z (ver Figura 46). Isto permite monitorizar o local da peça, ao longo da simulação, e assim fazer a distinção do tipo de peça montada.

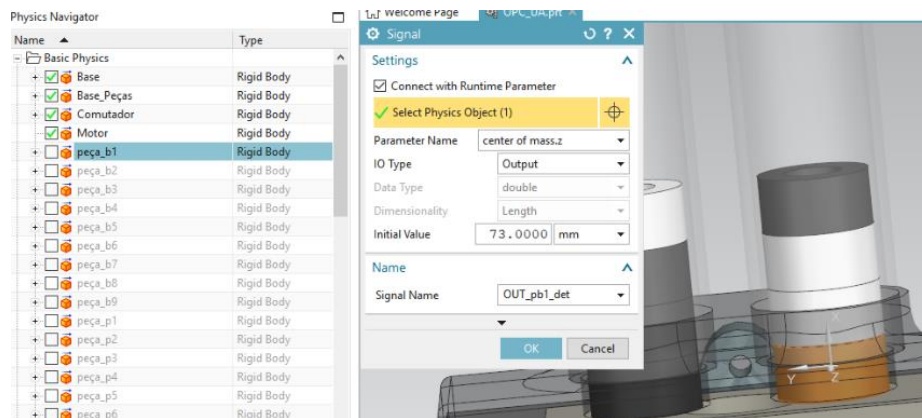


Figura 46 – Criação de sinais para deteção do tipo de peça

Para melhor perceber a troca de sinais efetuada, a Tabela 3, demonstra os sinais criados no *MCD* e a sua correspondência com o processo físico. A verde, estão assinaladas as saídas do *MCD* e os sensores do processo físico. A azul, as entradas do *MCD*, e os atuadores do *kit*. A coluna da tabela com a descrição, representa a definição de cada sinal, exceto nos sinais da deteção da cor das peças, pois foram realizados de forma diferente. Os principais sinais, estão descritos na próxima tabela, no entanto existem outros adicionados, que irão ser descritos posteriormente.

Sinais <i>MCD</i>	Sinais Processo Físico	Descrição
OUT_cnt	c	Centragem do posto
OUT_p2	p2	Sensor de posição 2
OUT_p1	p1	Sensor de posição 1
OUT_p0	p0	Sensor de posição 0
OUT_pup	a2	Peça presente superior
OUT_pdw	a1	Peça presente inferior
*OUT_det_pbn	c2	Cor da peça superior
*OUT_det_ppn	c1	Cor da peça inferior
IN_pb	Sb	Seleção peças brancas
IN_pp	Sp	Seleção peças pretas
IN_Mh	Mh	Roda em horário
IN_Mah	Mah	Roda em anti-horário
IN_VelF	F	Roda em modo rápido

Tabela 3 – Lista de sinais e correspondência

*Sinais criados para 4 peças de cada tipo, sendo a letra **n**, a que indica o número da peça.

Após a criação dos principais sinais, que permitem a interligação com dispositivos externos, outros procedimentos são necessários realizar, como as instruções. Estas instruções, permitem definir parâmetros de componentes do modelo, internamente no *MCD* e durante a simulação. É possível elaborar uma programação específica, para atribuir posições, velocidades, utilizar funções do *software*, entre outras opções. No modelo virtual criado neste projeto, as instruções elaboradas têm como objetivo afetar sinais de saída ou controlar atuadores, consoante os sinais criados para a interligação com uma comunicação externa.

Na Figura 47, verifica-se o exemplo de uma instrução, que através dos sinais recebidos por uma comunicação externa, é possível efetuar o controlo de velocidade da roda. Nesta imagem, os sinais recebidos para movimentar a roda, em sentido horário, anti-horário e velocidade rápida, são usados. O valor da velocidade rápida e da normal foram definidos, consoante várias observações do funcionamento do sistema real, após forçar esses movimentos e verificar tempos de processo. O valor positivo e negativo da velocidade, varia consoante o sentido que se movimenta a roda. Neste caso no sentido horário, movimenta-se com valor positivo, e no sentido anti-horário com valor negativo.

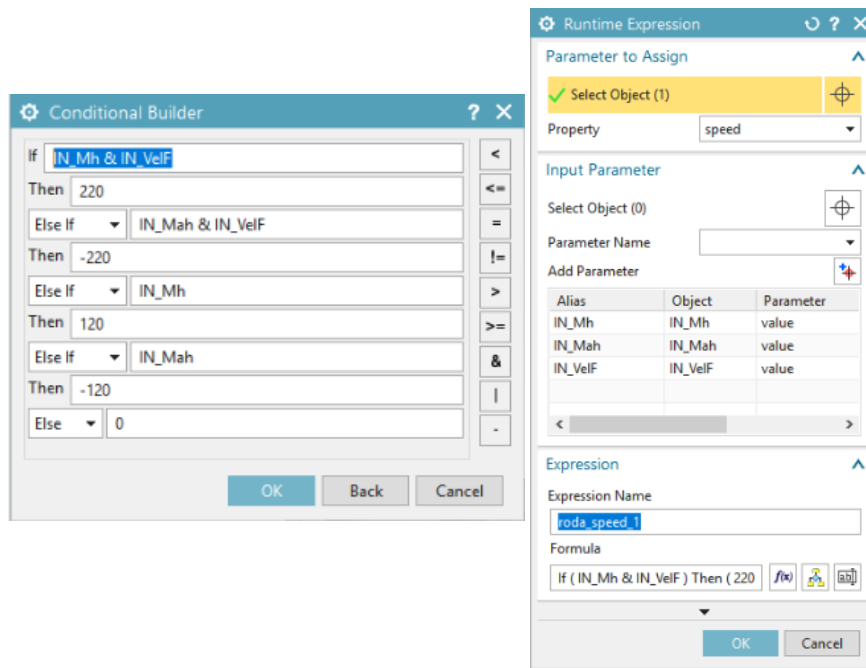


Figura 47 – Instrução de controlo de velocidade da roda

Ao contrário das operações, onde era efetuado o movimento da roda através da posição, inicialmente, foi utilizado, nesta fase mais avançada, o controlo por velocidade. Na realidade, o que é necessário é apenas definir a velocidade a que a roda se desloca, pois a posição é definida consoante os sensores existentes. A posição foi utilizada nas operações, como forma de testes iniciais e verificação de anomalias mecânicas.

Outra instrução adicionada, foi na condição de receber sinais, no que diz respeito ao comutador, onde é possível definir o controlo da posição, através destes. A maneira como foi definido esse controlo, é idêntica ao da roda, ou seja, se for recebido o sinal para fazer cair peças brancas este desloca-se para uma determinada posição e se for peças pretas para a posição contrária. Na movimentação para o centro, que permite colocar a peça por gravidade, no posto desejado, um outro sinal foi criado, **IN_pc**, para se movimentar no sentido dessa posição (posição 0 mm do comutador). No processo real, este último atuador não existe, sendo que quando é atuado o comutador para a seleção de peça branca ou preta, este retorna sempre ao centro, passado algum tempo. No modelo virtual, optou-se por elaborar um sinal que indica essa ordem, tornando mais simples o funcionamento. Na Figura 48 encontram-se os sinais utilizados para o controlo de posição do comutador.

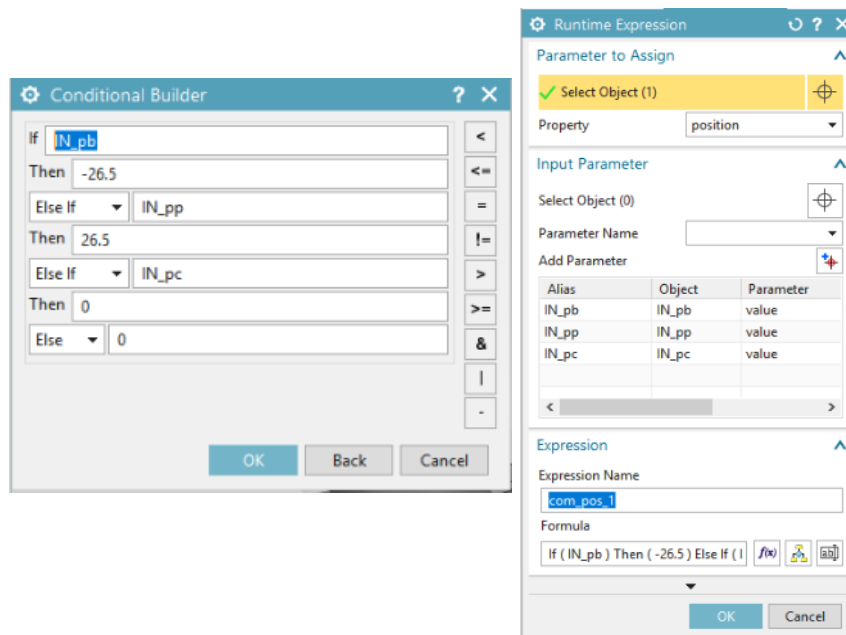


Figura 48 – Instrução de controlo de posição do comutador

No entanto, apenas o controlo de posição não é possível para fazer deslocar o comutador. É necessário associar a esse controlo, uma velocidade, como se pode ver na Figura 49.

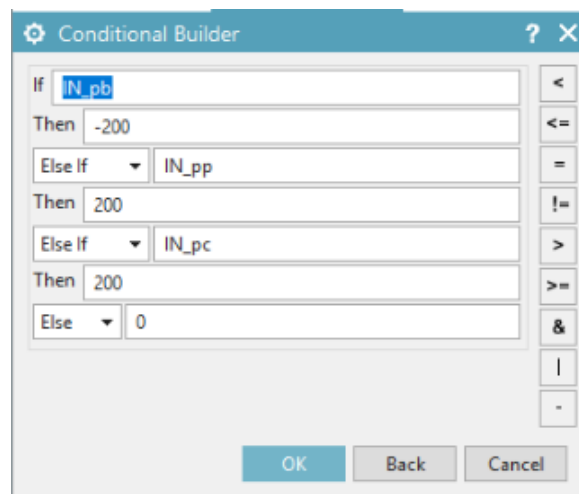


Figura 49 – Instrução de controlo de velocidade do comutador

Uma outra instrução foi criada, com o intuito de utilizar uma função interna do *MCD*, para obter o tempo de simulação decorrido, sendo aplicada para ativar um sinal de início de processo (**IN_start**). A Figura 50, demonstra a função utilizada. O sinal em descrição, fica ativo durante 100 milissegundos após a simulação iniciar.

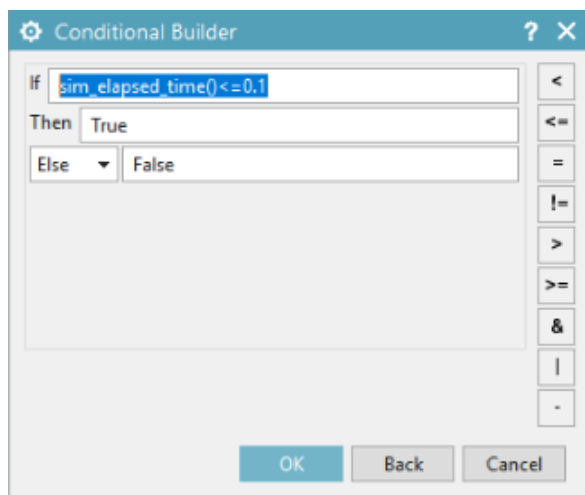


Figura 50 – Instrução de início da simulação.

Por fim, uma instrução, um sinal e um atuador foram criados para identificar a montagem de uma peça com cor errada. No *kit*, existe um atuador que é uma sirene, tendo como objetivo a sinalização de alarme no funcionamento deste. Neste caso, pode ser utilizado, para indicar que uma peça foi montada num posto, de forma errada. Para simular este alarme, um atuador de mudança de cor foi criado, elaborando um para cada sensor de peça inferior ou superior. A peça muda de cor para vermelho, devido à identificação da peça errada e quando as peças colidem com os sensores. Na Figura 51, encontra-se a criação do atuador.

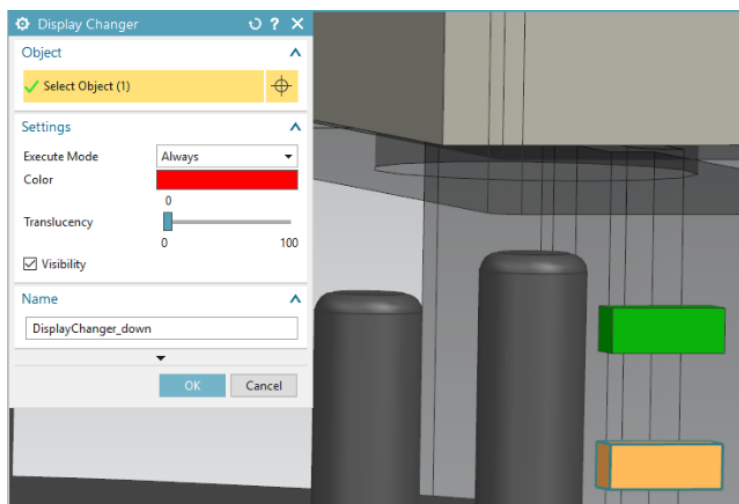


Figura 51 – Criação do atuador de mudança de cor da peça inferior

De seguida, um sinal foi criado (**IN_alm_cor**), idêntico ao do comutador (Figura 45), para o *MCD* receber informação do alarme, através do controlo externo, visto ser esse, a controlar a montagem das peças. Para concretizar a mudança de cor das peças, duas instruções foram adicionadas, para ativar esses dois atuadores. Na Figura 52, encontra-se o que foi descrito.

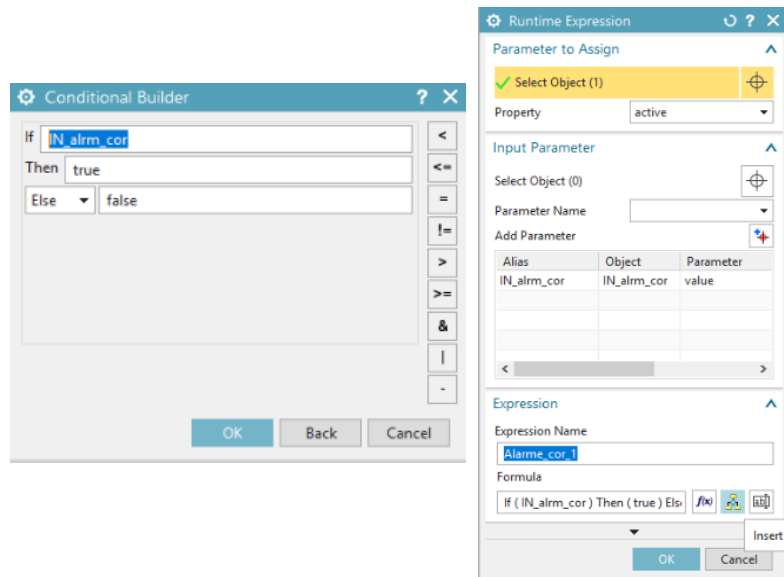


Figura 52 – Instrução de alarme de cor errada

3.3 Interligação com o *Digital Twin*

A interligação com o *DT*, permite a implementação do conceito de *VC*. Efetuar a simulação através de um controlo via *software* ou *hardware*, permite uma maior perceção do funcionamento de um processo físico. Com a utilização do *software MCD*, é possível efetuar uma interligação com um controlador externo, para efetuar essa simulação. Para além disso, se for necessário um controlo mais complexo do sistema, a interligação com um dispositivo externo é o ideal, porque torna mais fácil a programação dos movimentos a efetuar e a leitura de sensores.

O controlo de sistemas automáticos pode ser proveniente de vários dispositivos, que podem ser programados por *software* específico. Neste projeto, como já foi referido, foi aplicado um *PLC* da marca *Siemens*[®], cujo *software* utilizado para programação é o *TIA Portal V15*. O *PLC* vai ser o responsável pela lógica de programação do *DT*, que simula o processo físico existente.

Inicialmente, a interligação efetuada foi com o *PLCSIM*, uma aplicação da *Siemens*[®] que permite simular um *PLC* real. Nesta fase, uma simulação do tipo *SIL*, pode ser atingida, visto não ser necessário a execução de qualquer tipo de *hardware*, sendo tudo centrado só no *software MCD* e o *PLCSIM*. Nas outras interligações efetuadas, já foram implementados protocolos de comunicação utilizados na área da indústria, tais como o *TCP/IP*, *Profinet*, *OPC DA* e *OPC UA*. Com este método, é possível realizar uma simulação do tipo *HIL*, onde se testa o desempenho do *hardware*, ou seja, o *PLC* real com o *MCD*.

3.3.1 Programação do *Digital Twin*

A programação elaborada com o *TIA Portal V15*, permite a definição do *PLC* responsável pelo funcionamento do *DT*. Existem diversas séries de *PLCs* da marca *Siemens*[®], onde foram utilizados o *S7-1200* e o *S7-1500*. Estas séries estão disponíveis na *Siemens Automation Academy*, daí ser possível a sua utilização e conexão com o *TIA Portal*. A série *S7-1200*, é utilizada em processos automáticos na indústria, podendo criar redes de comunicação de *Profinet* e operar entradas e saídas, digitais e analógicas, no controlo desses (*S7-1200* na Figura 53) [84].



Figura 53 – PLC S7-1200 utilizado no projeto

Já o *S7-1500*, tem a possibilidade de executar as mesmas tarefas que o *S7-1200* e tem uma capacidade de desempenho maior. É um *PLC* mais avançado, que tem como objetivo ser aplicado em processos automáticos mais complexos e que necessitem de uma maior troca de informação (Figura 54) [85].



Figura 54 – PLC S7-1500 utilizado no projeto

Ambos os autómatos, permitem controlar sistemas automáticos numa rede *Profinet*, onde podem trocar dados com vários equipamentos, como *HMI*, *SCADA*, ilhas remotas etc. Contêm fichas *RJ45*, onde é possível conectar com estes, para programação e conexão com outros equipamentos, via *Profinet*, permitindo dados de transmissão, em tempo real.

Existem diversas linguagens de programação que podem ser aplicadas nos *PLCs*, para a execução de tarefas, onde as principais aplicadas neste projeto foram do tipo *Ladder* e *SCL*. A linguagem *Ladder*, é a linguagem de *PLC* mais utilizada, que pode ser descrita como uma linguagem de contactos, juntamente com blocos de função, originando em expressões de lógica. A linguagem *SCL*, tem a mesma característica, mas em vez de ser utilizada a linguagem de contactos, é usado o texto estruturado, aproximado à linguagem de programação *C* [86].

O *DT*, é composto por vários sensores e atuadores, logo, uma lógica de programação tem de ser efetuada para que todo o processo trabalhe de forma autónoma e eficiente. Sendo que, foi definida uma sequência de montagem para este sistema da mesa de montagem, consequentemente, um *GRAFCET* foi elaborado. Um *GRAFCET*, é utilizado na automação industrial e não só, pois é uma técnica, que permite o desenvolvimento de programas, que executam tarefas automáticas, de forma sequencial. Na Figura 55, encontra-se o *GRAFCET* elaborado de forma geral, para a sequência atribuída ao *DT*. Os sensores e atuadores descritos, são de acordo com os que se encontram na Tabela 3, e a sequência do *GRAFCET*, a da Tabela 2.

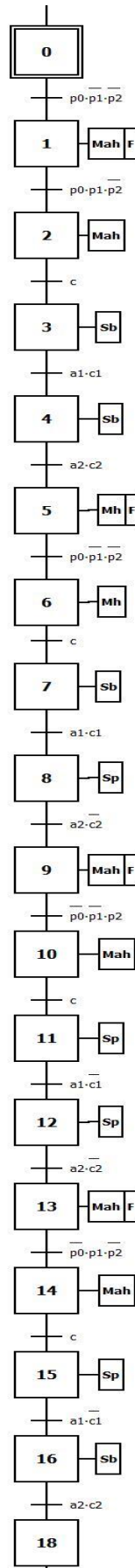


Figura 55 – GRAFCET de programação do DT

O *software TIA Portal*, para além da definição do autómato, tem outras secções importantes, utilizadas neste projeto, como a construção da rede de comunicação e os blocos de programação (Figura 56). Na construção das redes de comunicação, ao longo dos seguintes tópicos, vai se descrever a forma como se elaborou as várias redes de comunicação estudadas. No que toca aos blocos de programação, os *PLCs* utilizados, permitem definir vários tipos de blocos [87]:

- *OB (Organization Blocks)* – É a interface entre o sistema operacional e o programa do utilizador. A função *Main*, (função principal de um programa) por defeito, já se encontra criada pelo *software*, onde é possível chamar outras funções nesta e onde o *PLC* executa os ciclos do programa;
- *FC (Function)* – Funções onde se pode desenvolver lógica de programação, sem a possibilidade de armazenamento de dados, de forma cíclica. Foi usada para elaborar a lógica de programação do *DT* e a configuração de comunicações;
- *FB (Function Blocks)* – Idêntico às *FCs*, mas permitem o armazenamento de dados de forma cíclica através da utilização de *DBs*. São utilizadas em tarefas automáticas idênticas, mas associando a variáveis (*DBs*) distintas;
- *DB (Data Blocks)* – Criação de variáveis localizadas nas *DBs*, onde se podem utilizar em todo o programa.

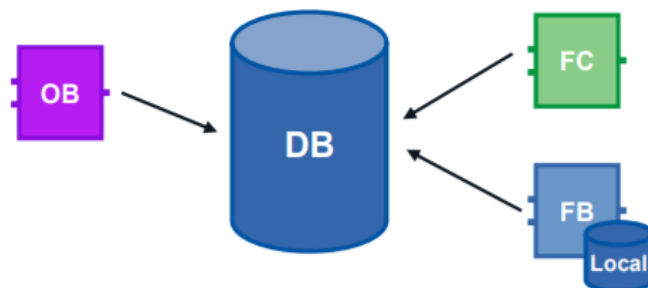


Figura 56 – Blocos de programação do *TIA Portal*

Tendo em conta os blocos de programação descritos, estes foram utilizados consoante as necessidades da lógica de programação aplicada. Como foram estudados vários meios de comunicação, a utilização destes blocos foi tida em consideração, de maneira a tornar *standard* o programa do *DT*. Ou seja, para cada modo de comunicação um novo programa teve de ser elaborado para parametrizar esse modo, no entanto, pretendeu-se aproveitar sempre a programação efetuada, relativamente ao funcionamento do *DT*. Esta programação *standard*, resulta do *GRAFCET* elaborado (Figura 55).

Um aspeto importante nos blocos de programação utilizados, é o facto de ter sido criada uma *DB*, com as variáveis a interligar com o *NX MCD* (Figura 57), que funciona de uma forma *standard*, pois é utilizada em todas as interligações. Pode se verificar que o nome das variáveis desta *DB* são exatamente o oposto aos da Tabela 3, pois neste caso, as entradas e saídas do *PLC*, são contrárias às do *NX MCD*. Neste caso, os sensores, são as entradas e os atuadores, as saídas, ao invés dos sinais criados no *software NX MCD*.

DB_MCD				
	Name	Data type	Start value	Monitor value
1	Static			
2	IN_cnt	Bool	false	FALSE
3	IN_p0	Bool	false	FALSE
4	IN_p1	Bool	false	FALSE
5	IN_p2	Bool	false	FALSE
6	IN_pdw	Bool	false	FALSE
7	IN_pup	Bool	false	FALSE
8	IN_start	Bool	false	FALSE
9	Out_Mh	Bool	false	FALSE
10	Out_Mah	Bool	false	FALSE
11	OUT_pb	Bool	false	FALSE
12	OUT_pp	Bool	false	FALSE
13	OUT_pc	Bool	false	TRUE
14	OUT_b1	Bool	false	FALSE
15	OUT_b2	Bool	false	FALSE
16	OUT_b3	Bool	false	FALSE
17	OUT_b4	Bool	false	FALSE
18	OUT_b5	Bool	false	FALSE
19	OUT_p1	Bool	false	FALSE
20	OUT_p2	Bool	false	FALSE
21	OUT_p3	Bool	false	FALSE
22	OUT_p4	Bool	false	FALSE
23	OUT_p5	Bool	false	FALSE
24	OUT_VeIF	Bool	false	FALSE
25	OUT_alm_cor	Bool	false	FALSE
26	IN_det_pb1	Real	0.0	0.0
27	IN_det_pb2	Real	0.0	0.0
28	IN_det_pb3	Real	0.0	0.0
29	IN_det_pb4	Real	0.0	0.0
30	IN_det_pp1	Real	0.0	0.0
31	IN_det_pp2	Real	0.0	0.0
32	IN_det_pp3	Real	0.0	0.0
33	IN_det_pp4	Real	0.0	0.0

Figura 57 – *DB* das variáveis do *DT*

Uma *FC*, foi criada para elaborar a programação do *GRAFCET* do processo. Como é um programa grande, decidiu-se colocar em anexo para demonstrar o processo elaborado. Esta *FC*, também é utilizada em todos os programas efetuados, para cada interligação, ou seja, é uma função *standard*, da programação efetuada.

3.3.2 PLCSIM

Com a elaboração da programação e criação de variáveis, procedeu-se à interligação com o *software NX MCD*, que simula o *DT* deste projeto. A interligação com a aplicação do *PLCSIM*, não necessita de qualquer tipo de *hardware*, sendo possível atingir o objetivo de simulação do tipo *SIL*. A versão de *PLCSIM* utilizada, foi a *S7-PLCSIM Advanced V2.0*, que pode ser iniciada como mostra a Figura 58. É necessário que o projeto elaborado no *TIA Portal*, esteja aberto, para que a aplicação do *PLCSIM*, detete o *PLC* criado no projeto. O nome do *PLC* deve ser introduzido no local adequado, efetuar o início da aplicação e colocar o *PLC* em modo *Run* (assinalado a verde).

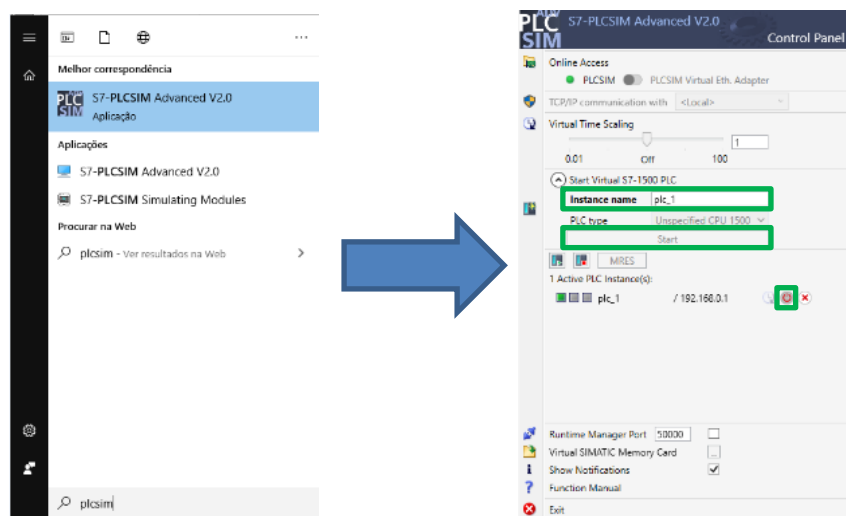


Figura 58 – Arranque do PLCSIM

Após este procedimento, um *PLC* virtual encontra-se criado e em modo *Run*, para executar as tarefas que lhe forem atribuídas. No projeto do *TIA Portal*, é necessário enviar o programa para este *PLC*, de modo a executar as tarefas automáticas do sistema. Na Figura 59, está demonstrado o ambiente do *TIA Portal*, onde é possível efetuar o envio do programa para o *PLC* (indicado a verde).

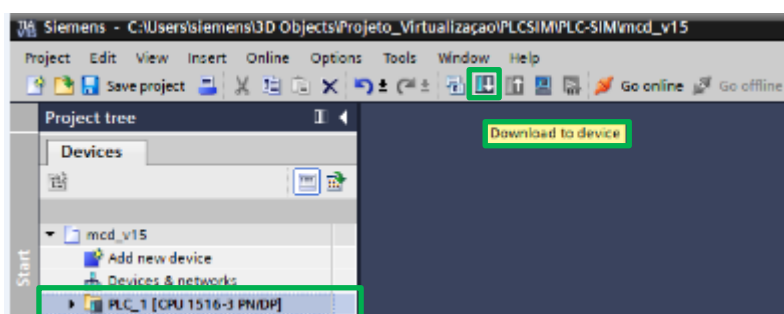


Figura 59 – Download para o PLC no TIA Portal

Estando os métodos efetuados no *TIA Portal*, o próximo passo é iniciar a interligação com o *NX MCD*. Para tal, é necessário iniciar este *software* e selecionar a configuração de sinais externos, como se verifica na Figura 60.

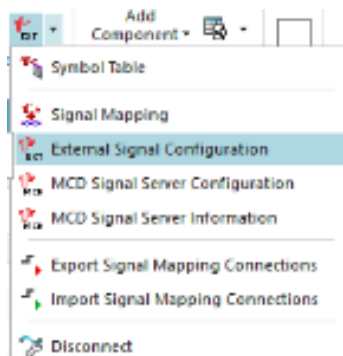


Figura 60 – Acesso a configuração de interligação externa no *NX MCD*

Ao selecionar esta opção, é possível selecionar os meios de comunicação, através de vários separadores, sendo que nesta situação, selecionou-se o separador do *PLCSIM Adv*. De acordo com a Figura 61, neste separador, existem vários parâmetros a serem definidos. O primeiro ponto a ser definido, é a instância criada pelo *PLCSIM*, que se vai interligar com o *NX MCD*. É necessário adicionar a instância em questão, ou efetuar a atualização da conexão, para que o estado desta se encontre em *Run*. Outra questão importante, é a escolha do tipo de memórias que se pretendem interligar com o *MCD*, como entradas (**I**), saídas (**O**) ou *DBs*. Por fim, e mais relevante, é a seleção das variáveis disponíveis nesta ligação, para depois interligar com os sinais criados no *MCD*. Estas variáveis aparecem automaticamente, ao efetuar a ligação.

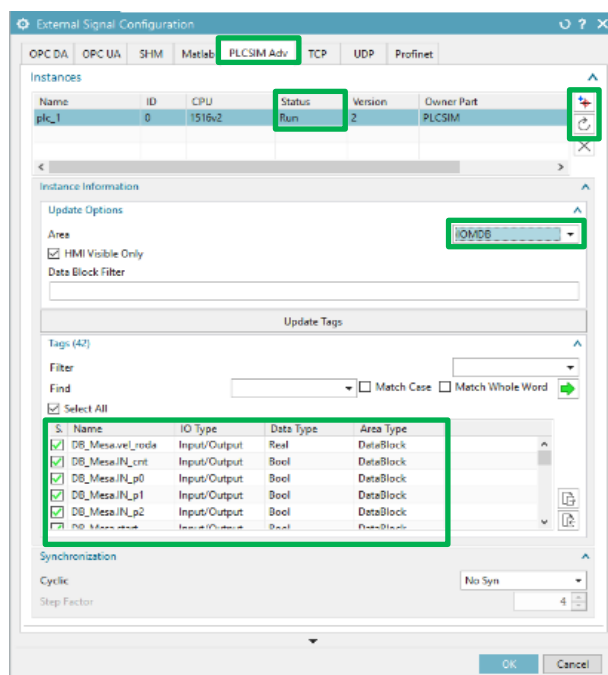


Figura 61 – Configuração da interligação *PLCSIM* no *NX MCD*

De seguida, procedeu-se à correspondência de sinais (*Signal Mapping*) entre o *PLC* e o *NX MCD*. O acesso a este menu, é da mesma forma que o acesso apresentado na Figura 60, onde mostra as várias opções na Figura 62. Neste menu de correspondência de sinais, no tipo de sinal externo, deve ser selecionado o *PLCSIM Adv*, e posteriormente o nome da instância, na qual é efetuada esta interligação. Na Figura 62, também se pode verificar que, a correspondência de cada sinal do *MCD* terá de ser efetuada com os sinais do *PLC*, para que os sensores e os atuadores funcionem de forma correta. Como exemplo, pode se verificar que o sinal **IN_Mh** do *MCD*, corresponde ao **OUT_Mh** do *PLC*, e o **OUT_p0** do *MCD* ao **IN_p0** do *PLC*. Respetivamente, estes sinais descritos, correspondem ao atuador de movimentação da roda no sentido horário, e a um dos sensores infravermelhos de posição. Todas estas tarefas, estão assinaladas a verde, na Figura 62.

Uma opção que é muito eficaz neste menu, mas não foi utilizada neste projeto, é a opção assinalada a vermelho. A opção permite efetuar a correspondência automática dos sinais de cada elemento, em vez de elaborar esta individualmente. Inicialmente, para que o nome das variáveis fosse perceptível, decidiu-se dar um acrónimo de **IN_** e **OUT_**, correspondendo a entradas e saídas de cada elemento, o *PLC* e o *MCD*. No entanto, poderia ter sido adotada esta técnica, apenas para um dos elementos, como por exemplo, o nome de um sinal de saída do *MCD*, **OUT_cnt**, poderia ter o mesmo nome do lado do *PLC*, mas, no *PLC*, iria ser uma variável de entrada, e não seria sucinta a sua denominação. Desta forma, era possível efetuar a correspondência automática (*Do Auto Mapping*), com a mesma designação de sinais.

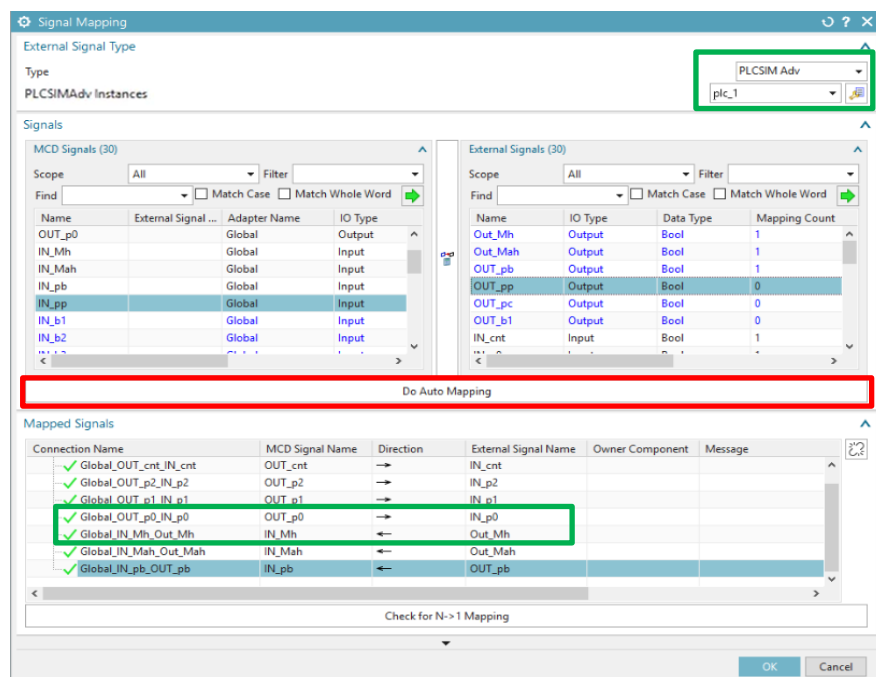


Figura 62 – Correspondência de sinais com *PLCSIM* no *NX MCD*

Com todos estes procedimentos elaborados, é possível iniciar a simulação através da interligação com o *PLCSIM*. Um dos primeiros problemas verificados com esta interligação externa, e que não foi possível observar com as operações internas do *MCD*, foi no facto de utilizar um sinal, para gerar peças como se verifica na Figura 40. Com as operações do *MCD*, é possível ativar o sinal que gera peças, previamente, para estas serem assembledas. No entanto, ao utilizar o mesmo sinal através do *PLCSIM*, foram realizadas várias formas de ativar este sinal, no *software* do *TIA Portal*, mas sempre sem sucesso. O acontecimento que reproduzia este problema, era que, em vez de serem ativas apenas as peças necessárias, um número enorme destas eram geradas, ou então nenhuma era criada. Isto provocava que a simulação entrasse em erro, ou ficava muito lenta, e as peças eram expulsas dos tubos como aconteceu anteriormente.

Um novo método foi realizado para corrigir este problema, descartando assim a fonte de objetos criada. O novo método, consiste em utilizar 4 peças de cada tipo, visto serem as necessárias para a execução da sequência atribuída. Um sinal para cada peça destas, foi criado, com o intuito de ativar esta, previamente, para ser assembleda no posto correspondente. Quando a peça é ativa, permite que esta se comporte fisicamente na simulação, pois caso não seja ativa, é como se fosse um objeto fixo. Na Figura 63, e com alguns ajustes mínimos necessários, foi possível obter sucesso no funcionamento desta interligação com o *DT*. Pode-se verificar que as últimas peças a serem assembledas, estão trocadas e devido a isso mudam de cor para vermelho.

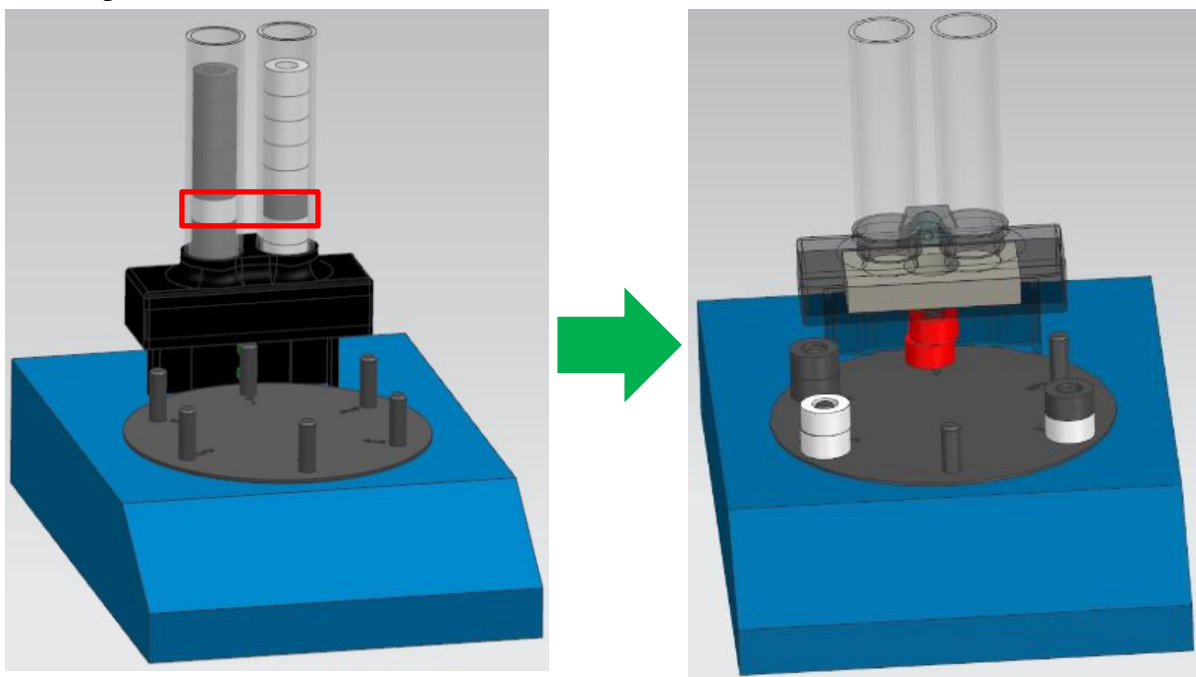


Figura 63 – Funcionamento interligação *PLCSIM*

3.3.3 TCP/IP

O protocolo *TCP/IP*, é o primeiro estudado com a implementação de um *PLC* real, neste projeto. A interligação de um *PLC* real por *TCP/IP*, com o *NX MCD*, ou outro elemento, requer de uma configuração de rede complexa no *TIA Portal*. Várias tentativas foram efetuadas para alcançar o sucesso da interligação com o *NX MCD* e muitas delas sem sucesso. Vários parâmetros têm de ser ajustados ao pormenor, para que seja possível a conexão.

Uma nova *FC* foi criada, para ser apenas dedicada a esta comunicação. Tendo em conta os autómatos usados e blocos de programação que o *TIA Portal* possui, é possível aplicar uma rede de *TCP/IP* entre o *PLC* e uma unidade que tenha a possibilidade de usar a mesma interface. Os blocos que o *TIA Portal* possui, são denominados por *TSEND_C* e *TRCV_C*, respetivamente, ao envio e receção de sinais, via *TCP/IP*. Relativamente ao bloco *TSEND_C*, é composto por várias entradas e saídas, onde têm as seguintes funções (Figura 64) [88]:

- *REQ* – Executa o envio de dados num pulso ascendente. Foi usado um oscilador de frequência que liga e desliga, juntamente com um contacto negado, da saída que indica que este está ocupado a enviar, e assim não entra em conflito;
- *CONT* – Controlo da comunicação. Sempre ligada para estabelecer conexão;
- *CONNECT* – Ponteiro para a estrutura da ligação. Criação de uma *DB* que é esse ponteiro;
- *DATA* – Ponteiro para a área de envio. Contém os endereços e o tamanho de dados a enviar;
- *BUSY* – Estado de ocupação do envio de dados. Se estiver ativo, não é possível enviar.

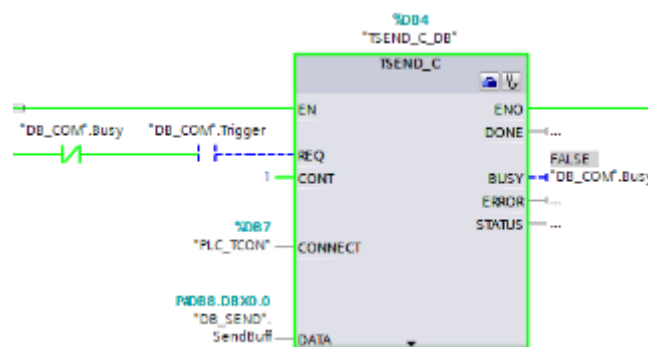


Figura 64 – Bloco *TSEND_C* no *TIA Portal*

De seguida, tem de ser configurado este bloco para definir o seu comportamento na rede de comunicação. Na Figura 65, observa-se que é necessário a definição do *IP*, do *PLC* e do *PC* que corre o *NX MCD*, que têm de ser da mesma família. O tipo de conexão é por *TCP* e também a parte ativa da conexão é importante definir (assinalado a verde).

A parte ativa neste caso, é o *NX MCD*, e a parte passiva, o *PLC*. Basicamente funciona um pouco como o *Client Server*, onde a parte ativa funciona como um *Client* e a passiva como um *Server*. O *NX MCD* executa a interação com serviços disponibilizados pelo *PLC*, mas neste caso via *TCP/IP*. A porta de ligação é importante colocar, para corresponder também no *NX MCD*.

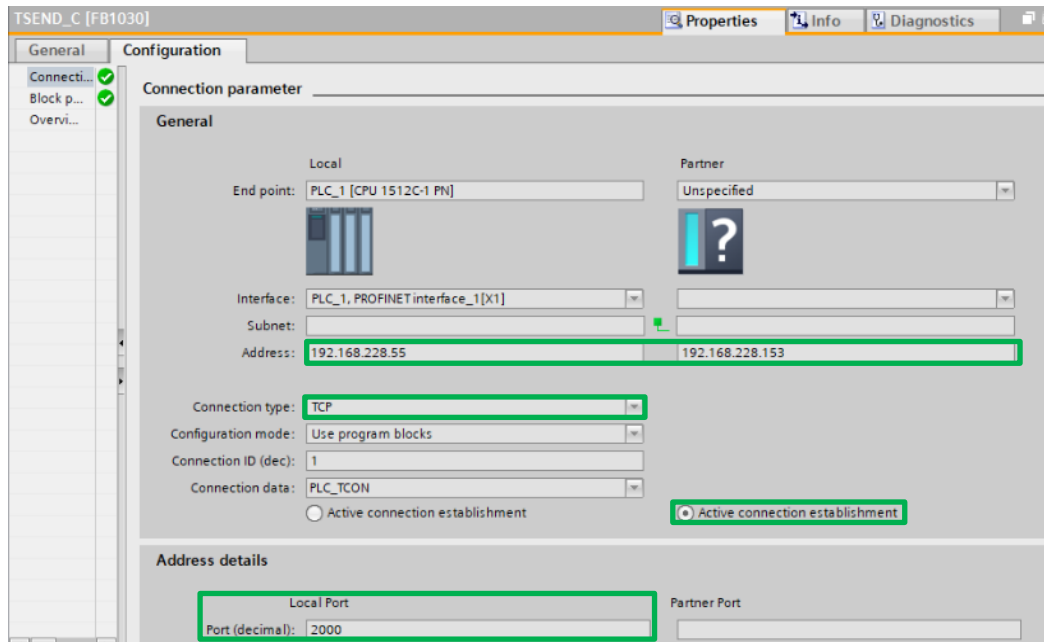


Figura 65 – Configuração do bloco *TSEND_C* no *TIA Portal*

Quanto ao bloco *TRCV_C*, este tem as seguintes características (Figura 66) [88]:

- *EN_R* – Habilitação para receber dados. Sempre ligado para permitir essa habilitação;
- *CONT* – Controlo da comunicação. Sempre ligada para estabelecer conexão;
- *LEN* – Número máximo de bytes a receber. Tamanho da *DB* que recebe dados do *NX MCD*;
- *CONNECT* – Ponteiro para a estrutura da ligação. Criação de uma *DB* que é esse ponteiro;
- *DATA* – Ponteiro para a área de receção. Contém os endereços e o tamanho de dados a receber;

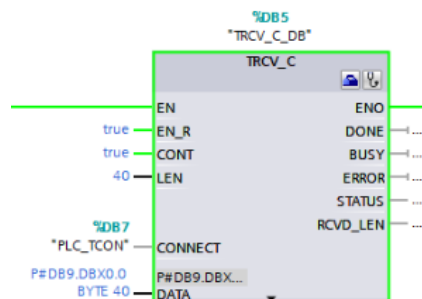


Figura 66 – Bloco *TRCV_C* no *TIA Portal*

Tal como foi elaborado no bloco *TSEND_C*, para o *TRCV_C* tem de ser configurada a parametrização da rede de comunicação. A configuração é exatamente idêntica à descrita na Figura 65, como se pode verificar na Figura 67 e assinalado a verde.

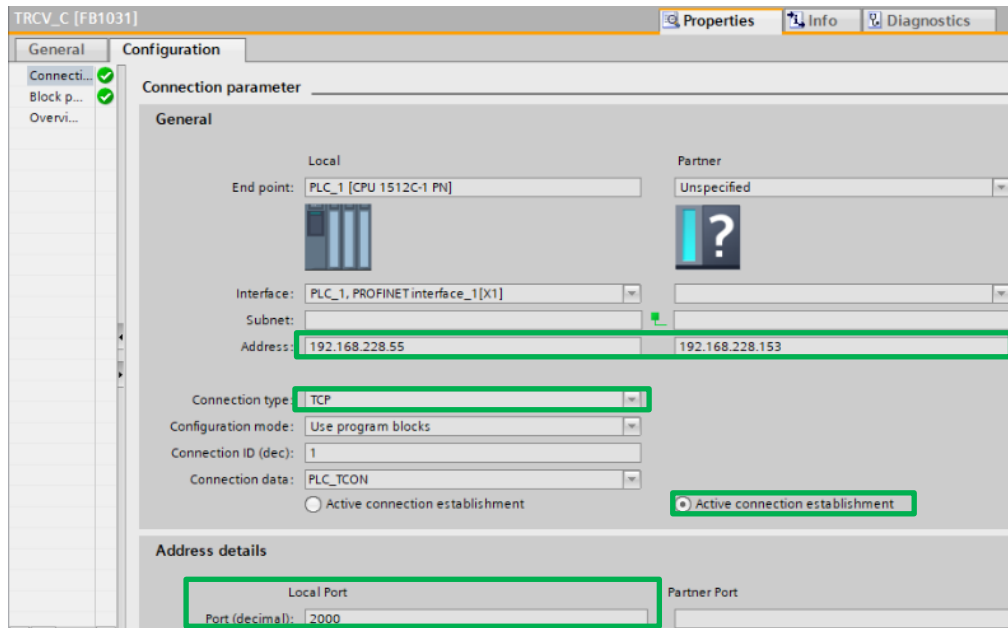


Figura 67 – Configuração do bloco *TRCV_C* no *TIA Portal*

Após a configuração da rede *TCP/IP*, através de blocos existentes no *TIA Portal* para o efeito, uma programação simples teve de ser adicionada, para o envio e receção de sinais, na *FC* nova criada. Esta programação, foi composta pela atribuição das variáveis da *DB standard* (Figura 57), às *DBs* criadas, para o envio e receção de sinais, inseridas nos blocos *TSEND_C* e *TRCV_C*. Na Figura 68, verifica-se essa atribuição. As *DBs* novas criadas, contêm endereços (assinalado a vermelho), para ser depois possível colocar no *NX MCD*, e corresponder de forma correta.

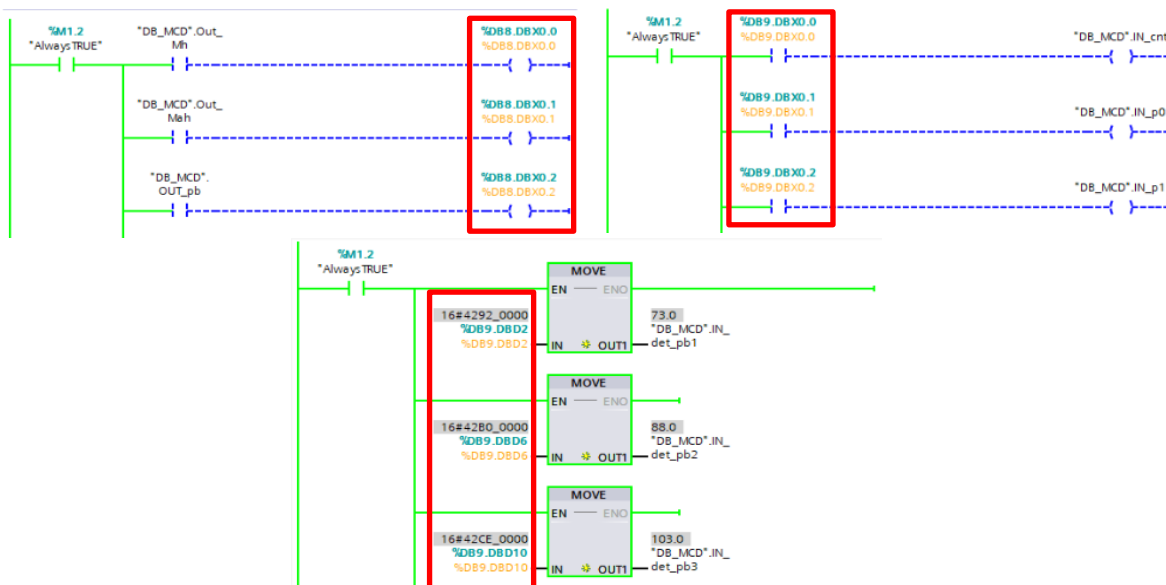


Figura 68 – Programação adicional para envio e receção de sinais

Para que estas *DBs* tenham os endereços, é necessário que não sejam otimizadas. É uma característica que estas têm, para armazenar variáveis de forma otimizada, e o *PLC* ter acesso, sem a necessidade de utilizar endereços [87]. No entanto, para que o *NX MCD* tenha acesso a estas variáveis é necessário aceder por endereço.

Na Figura 69, encontram-se as *DBs* criadas para a transmissão de sinais via *TCP/IP*. Basicamente, criou-se um vetor de *bytes* (*Array of Byte*), com o tamanho suficiente de variáveis a trocar com o *NX MCD*, pois torna-se mais simples trabalhar com esta unidade. Este tamanho tem de ser exatamente igual ao definido no *NX MCD*. Pode se verificar que a *DB* que recebe, tem o tamanho de 40 *bytes* e a que envia de 20. Cada *byte* permite alocar 8 *bits*, ou seja, o primeiro *byte* do vetor, pode ser utilizado para 8 variáveis booleanas, do endereço 0.0 ao 0.7. Já as variáveis reais utilizadas, ocupam 4 *bytes* na memória dos *PLCs* da *Siemens*®, podendo ser usadas do endereço 0 ao 3, por exemplo. Visto serem apenas usadas variáveis booleanas e reais, são estes os cuidados a ter em consideração, para definir *bytes* em variáveis distintas, e não alocar estas no mesmo espaço de memória.

DB_RCV			
	Name	Data type	Offset
1	Static		
2	ReceiveBuff	Array[0..39] ...	0.0
DB_SEND			
	Name	Data type	Offset
1	Static		
2	SendBuff	Array[0..19] ...	0.0

Figura 69 – *DBs* para receber e enviar sinais com o *NX MCD* via *TCP/IP*

Após a descrição do procedimento efetuado no *TIA Portal*, também é importante apresentar o mesmo para o *NX MCD*. Da mesma forma que foi elaborado, para a interligação com o *PLCSIM*, é necessário aceder ao mesmo menu e selecionar a interligação por *TCP* (Figura 70). Uma nova conexão tem de ser adicionada, colocando o *IP* e a porta. Depois tem de se atualizar a conexão, para que o estado desta se encontre acessível (*reachable*), tal como foi efetuado na interligação com o *PLCSIM*. O tempo de atualização deve ser tido em consideração também, pois permite que a interligação seja mais rápida e precisa.

Por fim, as variáveis a trocar com o *software* em questão devem ser inseridas, especialmente os tamanhos e os endereços, que devem coincidir exatamente, com o que foi realizado no *TIA Portal*. Após este processo, a correspondência de sinais deve ser efetuada selecionando para tal a ligação *TCP*, do mesmo modo que foi feito para o caso do *PLCSIM* (Figura 62). A simulação foi assim executada e verificou-se que tem o mesmo comportamento que na Figura 63, relativa à interligação por *PLCSIM*.

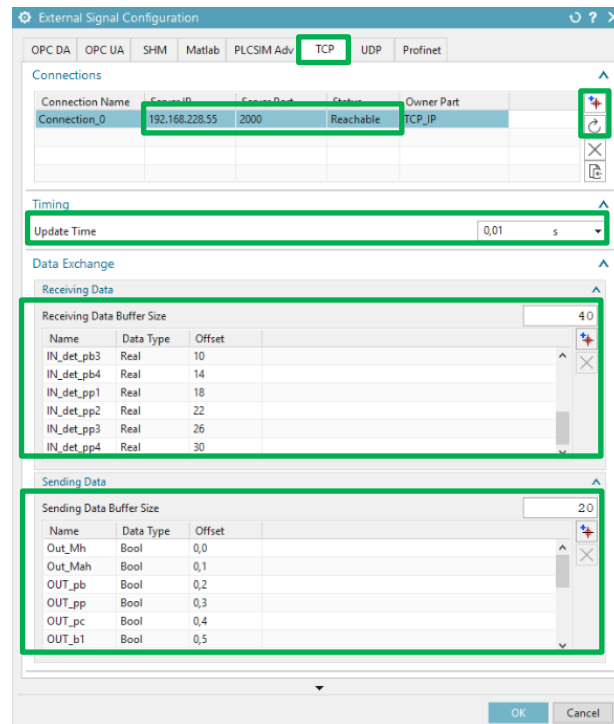


Figura 70 – Configuração da interligação TCP no NX MCD

3.3.4 Profinet

O protocolo *Profinet*, é aplicado na indústria, para interligar dispositivos de controlo, com a implementação de autómatos da *Siemens*[®]. Tal como no *TCP/IP*, na interligação com o *NX MCD*, é necessária uma configuração especial no *TIA Portal*. Para interligar o *PLC* com o *NX MCD*, via *Profinet*, é necessário utilizar a função de *IO device* e uma *PC station*. A *PC station*, representa um *PC* ou uma estação de operação, que inclui componentes da *Siemens*[®] como por exemplo, aplicações, como é o caso do *NX MCD*, onde este é configurado dentro da *PC station* [89]. Na Figura 71, pode se visualizar a criação de uma *PC station* no *TIA Portal* e os módulos a adicionar, para ser possível a conexão. Assinalado a verde, encontra-se a implementação de uma placa de rede *Ethernet* geral, que possibilita a ligação através da *Profinet*, e uma aplicação, que identifica o *NX MCD*.

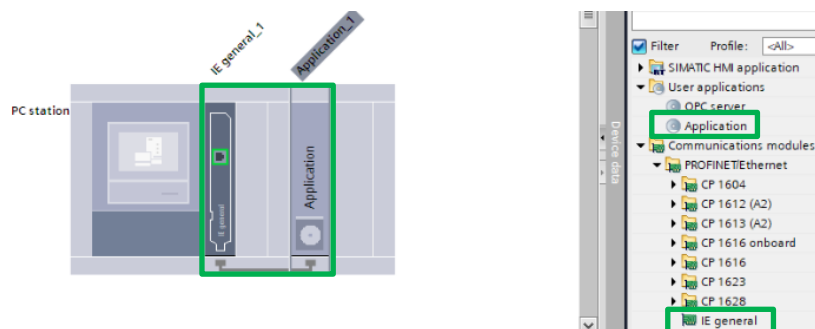


Figura 71 – Criação de PC station e módulos em Profinet no TIA Portal

Após a criação da *PC station*, é possível criar uma rede *Profinet* entre o *PLC* e esta, como se pode verificar na Figura 72. Para que esta comunicação seja possível, os *IPs* devem ser definidos de forma a serem da mesma família. O *IP* da *PC station*, deve ser o do computador onde corre o *NX MCD*. Neste caso apenas se usou um *PC*, para realizar todo o processo no *NX* e no *TIA Portal*.

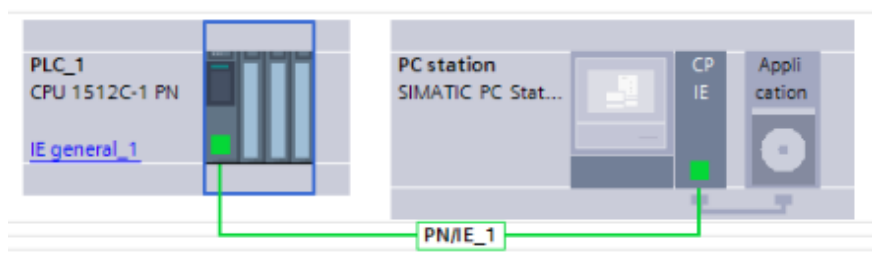


Figura 72 – Rede *Profinet* entre *PLC 1500* e uma *PC station* no *TIA Portal*

A função *IO device*, permite efetuar uma comunicação simples entre módulos da *Siemens*[®], *PLCs* ou ilhas remotas, por exemplo, em vez de usar métodos de comunicação, que necessitam de programação adicional, na interligação destes. Este método torna-se mais simples para trocar dados, pois é possível aceder a endereços de entradas e saídas, criadas para o efeito de comunicação *Profinet*, utilizando *IO device* [90].

Quanto à configuração no *TIA Portal*, esta apresenta-se na Figura 73, que demonstra o procedimento elaborado, para definir o *PLC* utilizado, como *IO device*. Para aceder a esta parametrização, é necessário aceder à configuração do dispositivo (*PLC*) e selecionar o modo de operação deste. Deve ser colocada a opção do *IO device*, para funcionar como tal, e selecionar o controlador com o qual se irá efetuar a comunicação, neste caso a *PC station*. Também deve ser selecionada, a opção que permite especificar, que a interface *Profinet* e as portas, são parametrizados pelo controlador *IO* de nível superior (*PC station*) [90]. Por fim, é necessária a criação das áreas de transferência, que correspondem aos endereços de entradas e saídas, e os tamanhos, que vão ser utilizadas para a troca de comunicação.

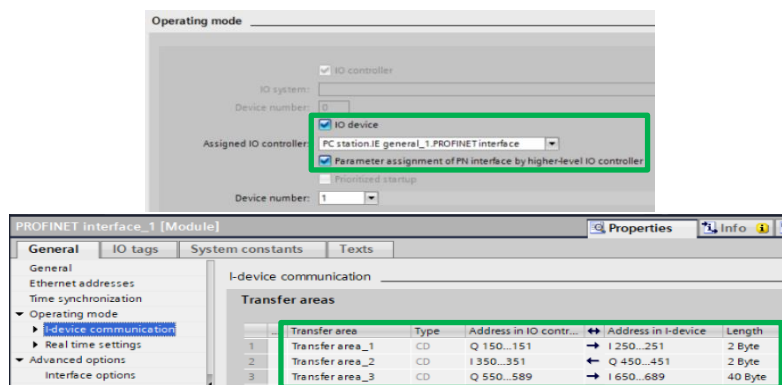


Figura 73 – Configuração de *PLC* para *IO device* no *TIA Portal*

Com este processo efetuado, também foi necessário implementar uma programação adicional numa *FC*, exatamente igual à descrita na Figura 68, tal como no protocolo *TCP/IP*. Neste caso, em vez de criar uma *DB* para fazer correspondência com a *DB standard*, foram criadas variáveis com endereços de entradas (**I**) e saídas (**Q**). Estes endereços, devem ser de acordo, com as áreas de transferência atribuídas, na configuração do *PLC* como *IO device*. A Figura 74, demonstra alguns endereços criados.

19	OUT_Mah	Default tag table	Bool	%Q450.1
20	OUT_Mh	Default tag table	Bool	%Q450.0
21	IN_det_pp4	Default tag table	Real	%ID678
22	IN_det_pp3	Default tag table	Real	%ID674
23	IN_det_pp2	Default tag table	Real	%ID670
24	IN_det_pp1	Default tag table	Real	%ID666
25	IN_det_pb4	Default tag table	Real	%ID662
26	IN_det_pb3	Default tag table	Real	%ID658
27	IN_det_pb2	Default tag table	Real	%ID654
28	IN_det_pb1	Default tag table	Real	%ID650
29	IN_start	Default tag table	Bool	%I250.6

Figura 74 – Variáveis de entrada e saída do *PLC*

Próximo passo, é utilizar a ferramenta *Station Configuration Editor*, que possibilita inserir módulos e componentes, selecionando os *slots* virtuais da *PC station*, e atribuir a sua funcionalidade. Com esta ferramenta, é possível executar a *PC station* de forma a estar operacional [91]. Para tornar mais fácil, e não correr o risco, de uma má configuração no *Station Configuration Editor*, o *TIA Portal* tem uma opção, para exportar um ficheiro com a extensão *XDB*, existente na configuração da *PC station*. Ao abrir esta ferramenta, é possível efetuar a importação desse ficheiro criado, onde é atribuída automaticamente, a configuração desta, como se pode ver na Figura 75. Deve se confirmar se no *Station Configuration Editor*, os parâmetros correspondem aos elaborados no *TIA Portal*, como o nome da estação e os módulos inseridos nos *slots*. Para a *PC station* funcionar, o modo deve estar em *RUN_P*. Os métodos descritos, estão assinalados a verde.

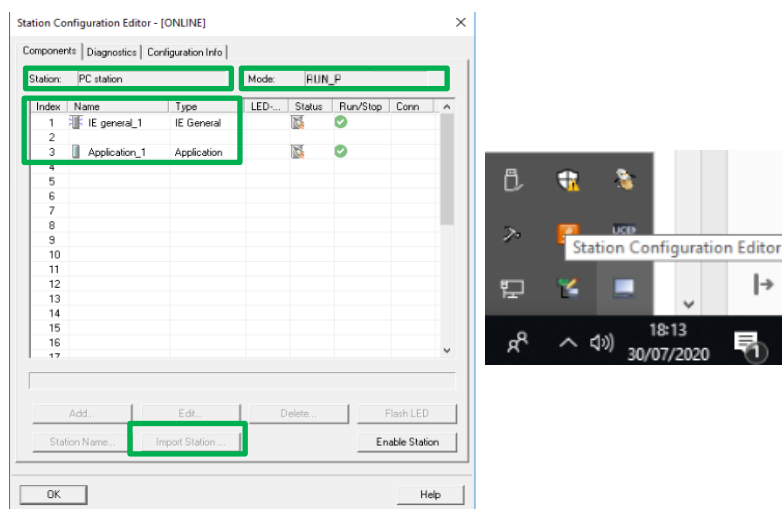


Figura 75 – Configuração do *Station Configuration Editor* com aplicação

Segue-se a configuração de sinais externa no *NX MCD*, com a opção da *Profinet*. Na Figura 76, pode se observar que neste protocolo de comunicação, é necessário adicionar as variáveis que se destinam à interligação, o seu tipo e o endereço. De notar, que os endereços devem corresponder às áreas de transferências realizadas na Figura 73. Para dar um exemplo, o endereço **I250.0**, que corresponde a uma entrada do *PLC*, deve ser respetivo ao endereço **Q150.0**, que é uma saída do *NX MCD*.

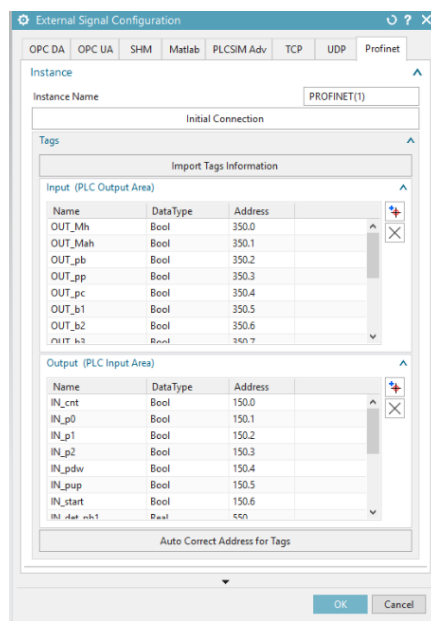


Figura 76 – Configuração de interligação *Profinet* no *NX MCD*

Por fim, deve se clicar no botão que inicia a conexão, e este irá apresentar o sucesso de ligação, se a *PC station* estiver bem configurada, como se pode verificar na Figura 77.

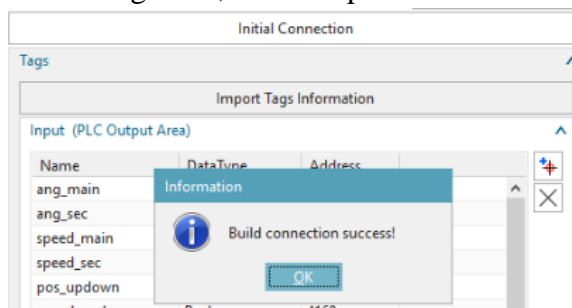


Figura 77 – Conexão inicial com *Profinet* no *NX MCD*

A interligação via *Profinet*, foi a única que não foi possível obter sucesso. Através de pesquisas, visualização de documentos e tutoriais, procedeu-se à realização de testes de interligação, mas todos sem sucesso. Os testes efetuados, consistiram na alteração de alguns parâmetros, como a configuração no *TIA Portal* com *IO Device*, os endereços de entradas e saídas, e a sua correspondência com o *NX MCD*. A simulação corre, mas a troca de sinais não é efetuada, por um possível lapso do *software* ou do *hardware*. Esta interligação, deverá ser mais estudada e aprofundada, pois existe pouca informação sobre esta.

3.3.5 OPC DA

Neste protocolo de comunicação, o procedimento utilizado no *software* do *TIA Portal*, é muito idêntico ao da *Profinet*. É necessário adicionar, para além do *PLC*, uma *PC station* que neste caso vai funcionar como um *OPC Server*, em vez de uma aplicação. Na Figura 78, está representada a criação da *PC station*, neste caso com um módulo de *OPC Server*. Ao adicionar este módulo, deve se ter em atenção a versão escolhida, pois a partir da versão 12, é utilizada apenas para outro tipo de conexão, como o *OPC UA*, e não é esse o objetivo neste tópico. Daí ser escolhida a versão 8.2, que permite interligar com sucesso este módulo [92].

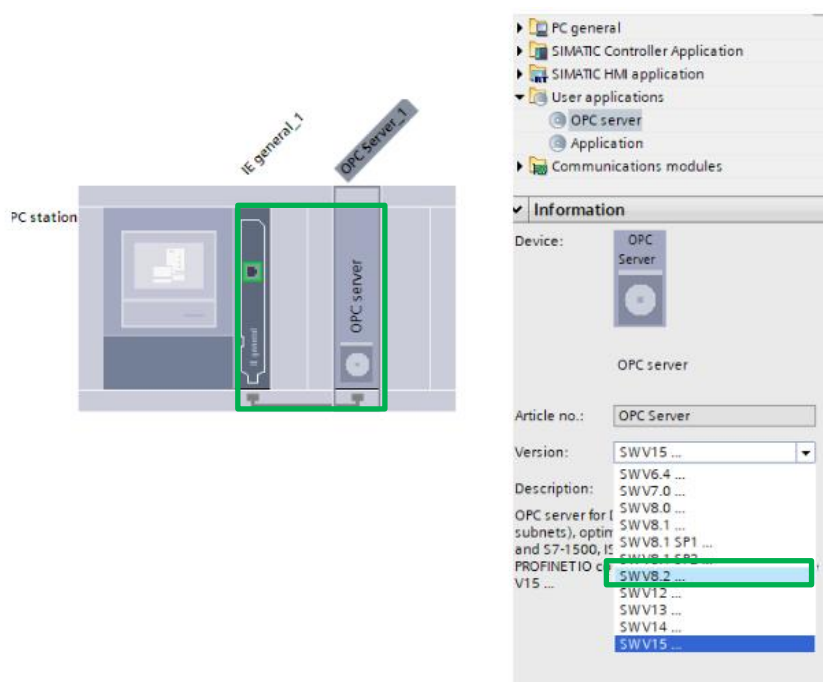


Figura 78 - Criação de *PC station* e módulos em *OPC DA* no *TIA Portal*

Após a criação da *PC station*, uma rede de comunicação tem de ser criada, como se fez na Figura 72. No entanto, para que seja possível a troca de informação entre a *PC station* e o *PLC* usado, é necessário adicionar uma conexão do tipo *S7 Connection*. O *OPC Server* estabelece este tipo de conexão, e o *S7-1200* vai ter um comportamento passivo. Este tipo de conexão, é utilizado para comunicar entre autómatos da *Siemens*[®], com a utilização da ligação *Profinet*. A Figura 79, demonstra várias imagens na criação de uma *S7 Connection*. Após a seleção da conexão (imagem de cima), deve se clicar no *OPC Server* e arrastar a ligação para o *PLC*. Ao arrastar esta ligação, irá aparecer a conexão em descrição, (imagem intermédia). Por fim, pode se verificar se a criação desta, foi elaborada com sucesso, de acordo com a última imagem [92]. Estes procedimentos, estão assinalados a verde e com setas pretas, nas imagens abaixo.

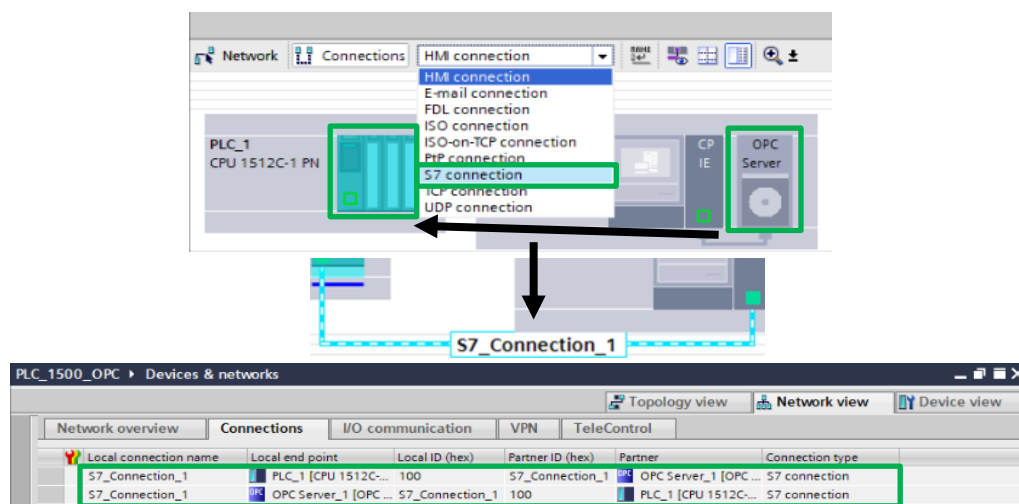


Figura 79 – Criação da S7 Connection para o OPC DA no TIA Portal

Tal como nas anteriores interligações com o *PLC* real, uma programação adicional, através da criação de uma *FC*, teve de ser implementada, onde nesta conexão, o método foi ligeiramente diferente. Devido a algumas limitações encontradas no *NX MCD* e no *OPC Scout*, que vão ser aprofundadas mais abaixo, foi necessário definir variáveis do tipo **M**. As variáveis deste tipo são memórias internas, de *PLCs* da *Siemens*[®], como o *S7-1200* e *S7-1500*, utilizadas com o mesmo efeito das *DBs*, mas neste caso têm sempre endereços associados. Nas variáveis de entrada do *PLC*, tanto do tipo booleana, como real, o método elaborado foi idêntico aos anteriores. Apenas foram criadas variáveis do tipo inteiro, nas saídas do *PLC*, para corresponder às variáveis do tipo booleanas. Por exemplo, **MW10**, toma o valor de 1, se a variável for verdadeira, e no caso de falsa, fica com valor de 0 (ver Figura 80).

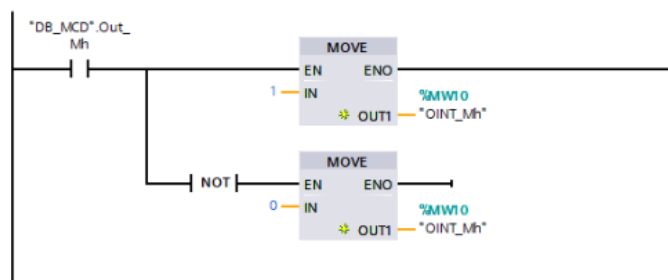


Figura 80 – Programação adicional com OPC DA no TIA Portal

Com estes passos efetuados, segue-se a inicialização do *OPC Server* no *software OPC Scout*. O *OPC Scout*, é um *software* que permite estabelecer a ligação com um servidor *OPC*, acedendo a dados do *PLC*, visualizando e forçando variáveis, para observar o estado da conexão. Primeiro que tudo, deve se configurar a ferramenta *Station Configuration Editor*, de maneira a ter o mesmo aspeto que o da Figura 81 (assinalado a verde). Para tornar mais simples, o procedimento foi igual ao efetuado anteriormente, para a interligação via *Profinet*, onde é possível exportar um ficheiro do *TIA Portal*, e depois importá-lo nesta ferramenta.

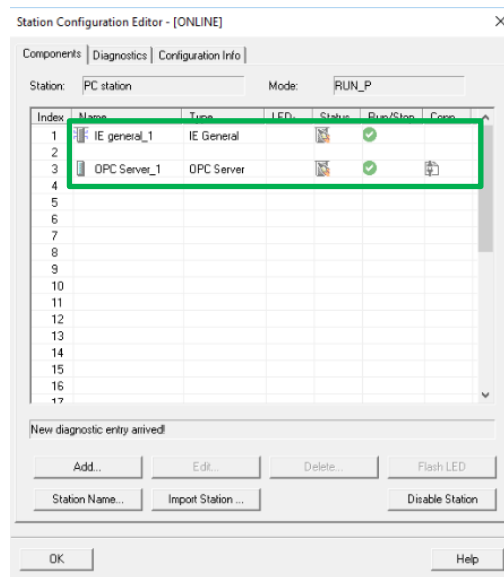


Figura 81 - Configuração do *Station Configuration Editor* com *OPC Server*

Com esta configuração, pode se iniciar o *software OPC Scout*, que executa a conexão do *OPC Server*. Para estabelecer a ligação, deve-se selecionar o *Server* com a descrição *OPC.SimaticNET*, como se observa na Figura 82, assinalado a verde [92].

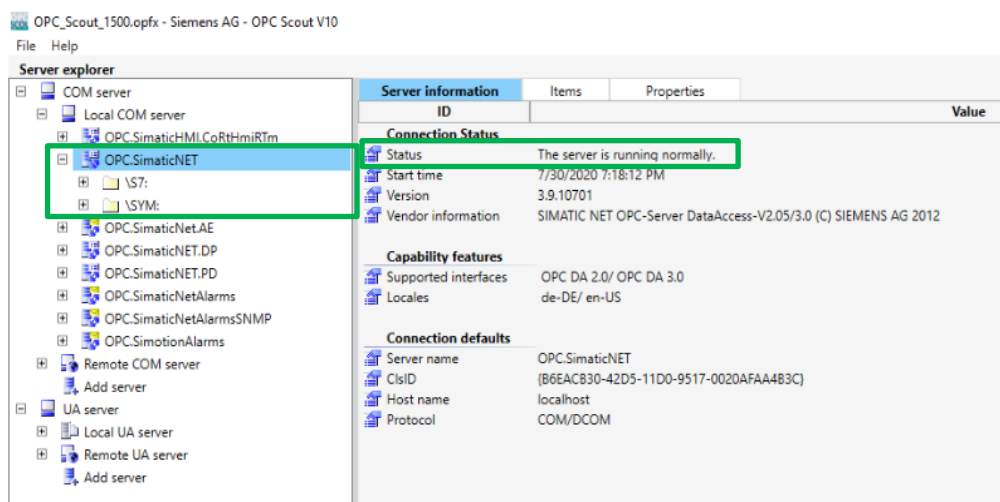


Figura 82 – Conexão com o *Server OPCDA* no *OPC Scout*

Estando a ligação estabelecida, segue-se as definições de variáveis com as quais se pretende comunicar com o *PLC*. Para localizar as variáveis de ligação com o *PLC*, tem-se neste *OPC Server*, a *S7 Connection* criada, onde podem ser definidas as variáveis, na secção dos objetos desta conexão, como mostra na Figura 83 a verde. No *OPC Scout*, não foi possível definir variáveis do tipo *DB* (assinalado na Figura 83 a vermelho), pelo que se optou pelas do tipo *M*. Daí ter sido usado este tipo de memórias, na programação adicional.

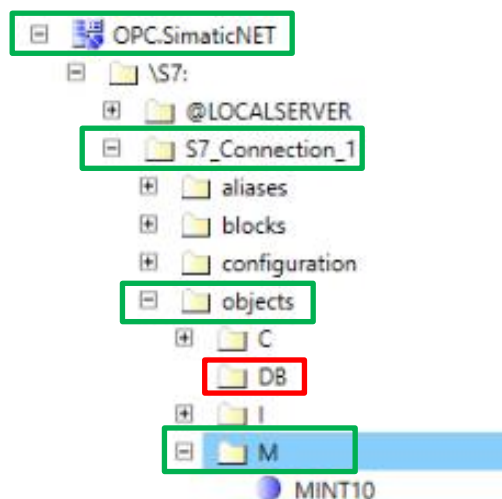


Figura 83 – Acesso a variáveis do *OPC Server* no *OPC Scout*

Na criação de variáveis, tem de se ter em atenção alguns aspetos. Esses aspetos são o tipo de variável, o endereço, o bit associado, e o número de valores, que corresponde ao tamanho da variável. Na Figura 84, está exemplificada a criação de uma variável do tipo booleana, mais concretamente a **MX8.0**.

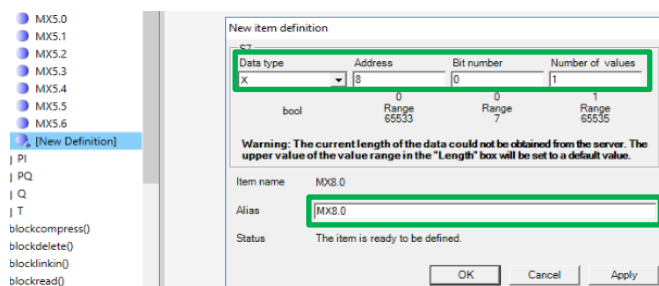


Figura 84 - Criação de variáveis no *OPC Scout*

Ao iniciar o *NX*, é preciso aceder à configuração de sinais externa e seleccionar o separador *OPC DA*. Neste separador, é necessário adicionar o *OPC DA Server*, conectado anteriormente através do *OPC Scout* (*OPC.SimaticNET*), e atualizar o seu estado de ligação, verificando que está conectado. Após estes passos, o tempo de atualização também deve ser ajustado, para obter uma interligação mais rápida. A Figura 85, demonstra o sucedido (assinalado a verde).

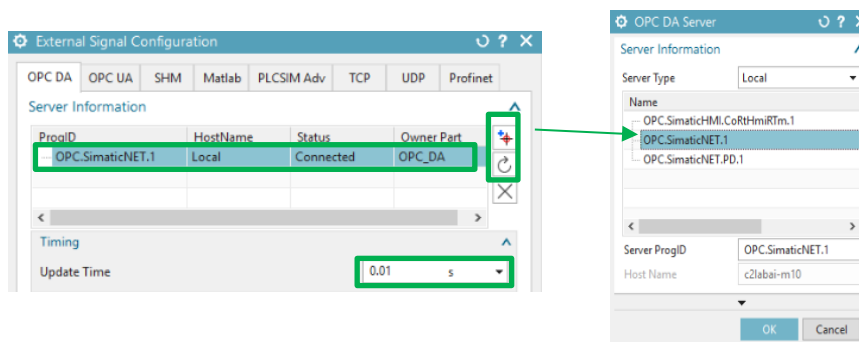


Figura 85 – Ligação com o *OPC DA Server* no *NX MCD*

De seguida, é necessário procurar as variáveis definidas no *OPC Scout*, que são acessíveis, da mesma forma como foram criadas neste. Ou seja, deve se expandir o *OPC Server*, seleccionar a *S7 Connection*, aceder aos objetos e verificar as variáveis criadas do tipo **M**. Devem ser seleccionadas, as que se pretendem utilizar para a interligação com o *PLC* (Figura 86).

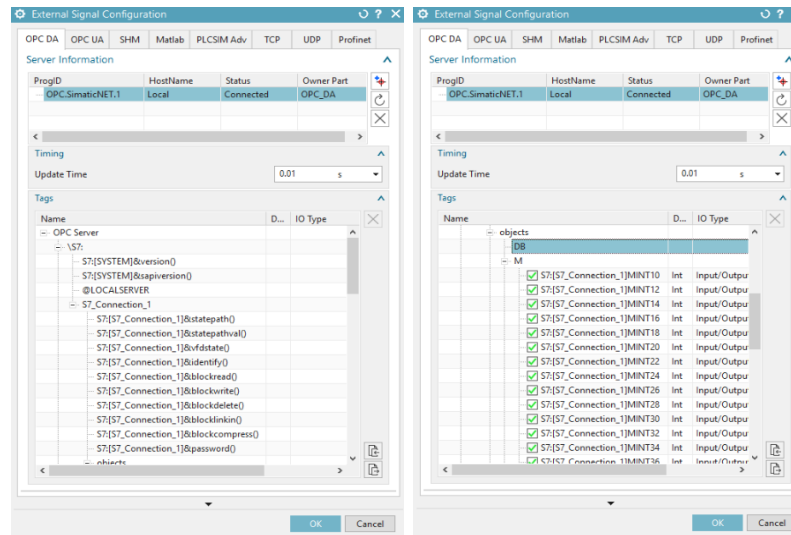


Figura 86 – Seleção de variáveis a interligar no *NX MCD* com *OPC DA*

Na atribuição das saídas do *NX*, a Figura 87, mostra que nas variáveis booleanas, só o bit é que é acessível (indicado a verde). Isto significa que se forem criadas, por exemplo, as memórias **M5.0** e **M6.0**, só a variável com o nome 0, é que aparece no *NX*, não existindo distinção entre essas duas memórias criadas. Optou-se então, por manter as saídas do *NX* como booleanas, pois são apenas 7 memórias, da **M5.0** à **M5.6**. Visto que nos *PLCs* da *Siemens*[®], cada espaço de memória tem 8 *bits* (**MB5**), significa que se pode utilizar da memória **M5.0** à **M5.7**. Assim sendo, novos sinais e instruções tiveram de ser adaptadas, para as variáveis de entrada do *NX MCD*, do tipo inteiro. É necessária a atribuição de valores inteiros a booleanos, ou seja, se tiver o valor de 1, significa que o sinal está verdadeiro, e se receber o valor de 0, está a falso.

Name	D...	IO Type
<input checked="" type="checkbox"/> S7[SYSTEM]&version()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&majorversion()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&minorversion()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&revision()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&buildnumber()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productname()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productversion()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&producttype()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productfamily()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productline()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productmodel()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productvariant()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption2()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption3()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption4()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption5()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption6()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption7()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption8()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption9()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption10()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption11()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption12()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption13()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption14()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption15()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption16()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption17()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption18()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption19()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption20()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption21()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption22()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption23()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption24()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption25()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption26()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption27()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption28()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption29()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption30()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption31()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption32()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption33()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption34()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption35()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption36()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption37()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption38()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption39()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption40()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption41()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption42()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption43()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption44()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption45()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption46()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption47()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption48()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption49()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption50()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption51()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption52()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption53()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption54()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption55()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption56()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption57()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption58()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption59()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption60()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption61()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption62()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption63()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption64()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption65()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption66()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption67()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption68()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption69()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption70()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption71()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption72()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption73()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption74()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption75()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption76()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption77()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption78()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption79()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption80()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption81()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption82()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption83()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption84()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption85()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption86()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption87()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption88()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption89()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption90()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption91()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption92()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption93()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption94()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption95()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption96()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption97()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption98()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption99()	Real	Input/Output
<input checked="" type="checkbox"/> S7[SYSTEM]&productoption100()	Real	Input/Output

Figura 87 – Variáveis booleanas com *OPC DA* no *NX MCD*

Por fim, deve ser efetuada a correspondência de sinais, seleccionando a ligação *OPC DA*, tal como foi feito com as interligações anteriores. A simulação foi iniciada com sucesso e verificou-se que tem o mesmo comportamento que o das interligações por *TCP/IP* e por *PLCSIM*.

3.3.6 OPC UA

O protocolo *OPC UA*, foi o último e o mais importante protocolo estudado neste projeto, pois é dos protocolos mais explorados na Indústria 4.0, e na qual se centra, o tema deste projeto. Relativamente aos protocolos utilizados com o *PLC* real, este é o que tem a configuração mais simples de todos, pois o próprio *PLC* permite ativar um *OPC UA Server*, e no *NX MCD*, é possível aceder a vários elementos. Iniciando a configuração no *TIA Portal*, na secção da configuração de *hardware* do *PLC*, é possível ativar um *OPC UA Server*, que o próprio *PLC* executa. Para além disso, também se pode verificar o endereço do *Server*, no qual é possível o seu acesso, a identificação da porta, e os tempos mínimos de amostragem e publicação. Nestes tempos, o mínimo possível a definir, é cerca de 0.1 segundos e 0.5 segundos, respetivamente. Na Figura 88, está demonstrado o procedimento descrito.

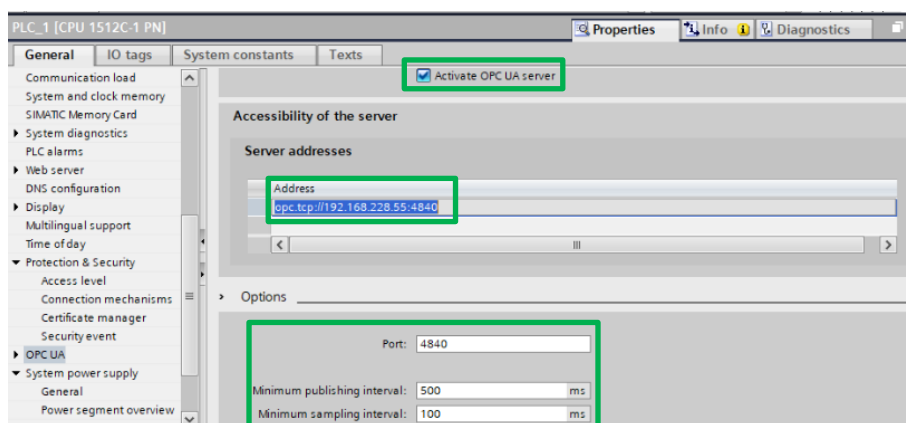


Figura 88 – Configuração do PLC com OPC UA no TIA Portal

Também é importante não selecionar qualquer tipo de segurança, no que toca ao acesso a este *Server*, para não existir limitações na troca de dados. A segurança deste *Server* pode ser utilizada, quando é necessário existir algum tipo de proteção dos dados, para prevenir acessos proibidos, a pessoas não autorizadas. Na Figura 89, está representada a segurança do *Server OPC UA* utilizado.

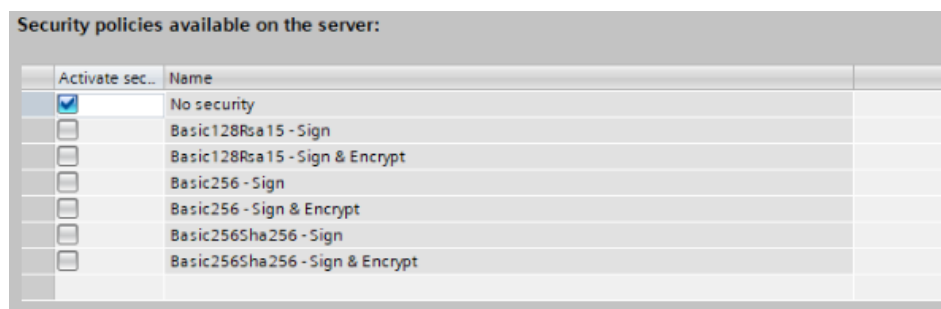


Figura 89 – Segurança do Server OPC UA no TIA Portal

Apenas são necessários estes passos no *TIA Portal*, para realizar esta interligação, não sendo necessário nenhuma programação adicional, como se fez nas outras interligações com o *PLC* real. Segue-se então o procedimento no *NX MCD*, para aceder ao *Server OPC UA*, onde deve ser seleccionada, na configuração de sinais externas, a interligação *OPC UA*. Tal como na interligação por *OPC DA*, também é possível adicionar um *Server OPC UA*, e atualizar o seu estado, para estabelecer a conexão com este. Ao adicionar o *Server*, deve se copiar o endereço obtido pelo *TIA Portal*, e colocar no *NX MCD*. O tempo de atualização neste caso deve ser de 0.1 segundos, devido ao tempo de amostragem do *OPC UA Server*, definido no *PLC*. A Figura 90 comprova os aspetos definidos no *NX MCD*, indicados a verde.

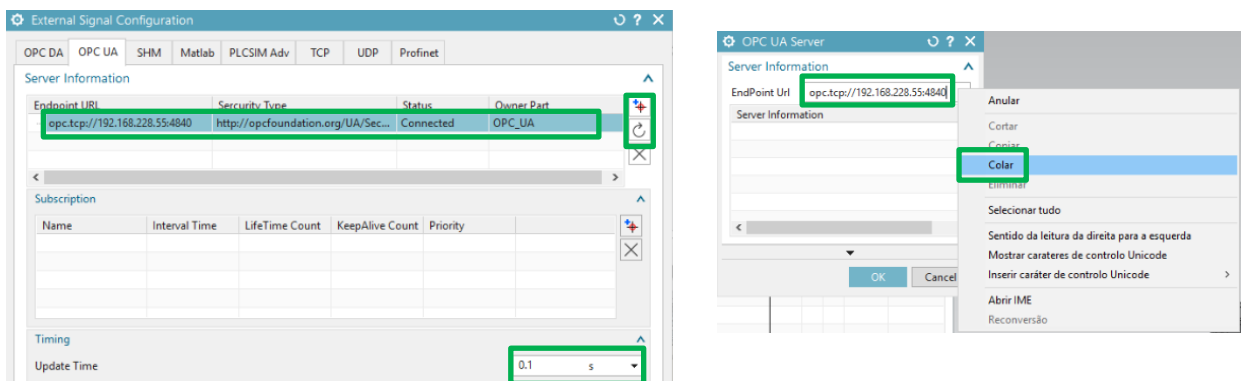


Figura 90 – Ligação com *Server OPC UA* no *NX MCD*

De seguida, devem ser seleccionadas as variáveis que se pretendem utilizar para a interligação, como se pode observar na Figura 91. Neste protocolo, é possível aceder à *DB standard* criada, logo o acesso é direto como por *PLCSIM*, e daí, também não ser necessária nenhuma programação adicional no *TIA Portal*.

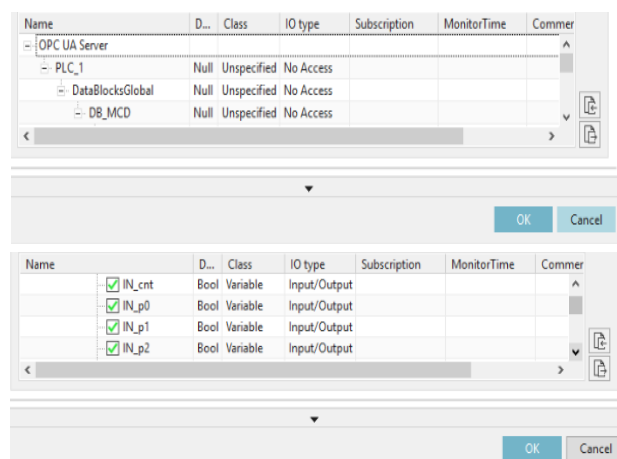


Figura 91 – Seleção de variáveis a interligar no *NX MCD* com *OPC UA*

A interligação foi realizada com sucesso, após a seleção de variáveis, para coincidirem. A simulação tem um comportamento idêntico aos anteriores, notando apenas uma ligeira diferença, no tempo de atualização do estado das variáveis, que é um pouco mais lento.

4. Resultados e problemas

Neste tópico, vão ser descritos os resultados obtidos e problemas encontrados, tendo em conta o trabalho desenvolvido neste projeto, principalmente, sobre o *software* do *TIA Portal* e do *NX MCD*. A implementação do *DT* abrange várias vertentes, como o desenho do modelo digital, a programação, interligação e a simulação de movimentos, para efetuar uma cópia virtual do processo real.

4.1 Versões de *software* e *firmware*

Um dos desafios encontrados, foi o facto das versões do *software* utilizado e do *firmware* dos *PLCs*, não serem os mais adequados, para certas tarefas de interligação entre o *TIA Portal* e o *NX MCD*. Tendo em conta os recursos aplicados neste projeto, que se encontram na *Siemens Automation Academy*, tanto a nível de *hardware* como *software*, existiam algumas limitações no início do projeto, para elaborar as interligações pretendidas. Para resolver tal questão, foi necessário atualizar o *firmware* do *PLC S7-1500*, bem como o *software NX MCD*. Em relação à gama do *PLC S7-1200* existente no laboratório, este não permite atualizar o *firmware* para além da versão 3.0. Quanto à versão do *software TIA Portal* utilizada, a versão foi a 15, onde é possível adicionar as versões de *PLCs* corretas e interligar com o *NX*. Com as interligações efetuadas, e tendo em atenção o comportamento das simulações, a próxima tabela, demonstra as versões dos vários recursos utilizados, para cada interligação, e a necessidade de cada um possuir essa particularidade.

	<i>FIRMWARE</i>		<i>SOFTWARE</i>		
Interligação	<i>S7-1200</i>	<i>S7-1500</i>	<i>NX MCD</i>	<i>TIA Portal</i>	Estado
<i>PLCSIM</i>	V3.0	V1.8 ou V2.0	V12.0 ou V1915	V15	Ok
<i>TCP/IP</i>	V3.0	V1.8 ou V2.0	V12.0 ou V1915	V15	Ok
<i>PROFINET</i>	V4.0	V1.8 ou V2.0	V12.0 ou V1915	V15	Nok*
<i>OPC DA</i>	V3.0	V1.8 ou V2.0	V12.0 ou V1915	V15	Ok
<i>OPC UA</i>	V4.4	V2.0	V1915	V15	Ok**

Tabela 4 – Lista de versões compatíveis entre *software* e *firmware*

* - Interligação só possível com o *S7-1500*, sem sucesso.

** - Interligação só possível com o *S7-1500*, com sucesso.

4.2 Resultados e problemas na simulação

Durante o decorrer do projeto, várias questões surgiram, que levaram a que alguns problemas surgissem e causassem restrições, no desempenho da interligação entre o *PLC* e o *MCD*. Começando pelo facto, da simulação de movimentos no *MCD*, ter um comportamento diferente em certos aspetos. Se a simulação decorrer apenas com o *software MCD*, como foi descrito no tópico 3.2.3, o comportamento dos movimentos mecânicos do modelo, é sempre o mesmo e o resultado é o esperado. No entanto, ao existir uma interligação com um controlo externo, cada vez que é iniciada a simulação, esta pode ter um comportamento diferente. O próprio *software* ou o computador, podem estar limitados de alguma forma, através da conexão com um controlo externo. Isto leva a que o funcionamento do *DT*, nem sempre corresponda da mesma forma que o processo real, ficando por vezes mais lento e inconstante.

Alguns exemplos dos problemas encontrados derivados desta situação, são o facto dos movimentos mecânicos serem menos precisos, ou os tempos utilizados na programação do *PLC*, não corresponderem ao tempo pretendido, pois a simulação torna-se mais lenta. Como é imprevisível esta situação, ao correr a simulação, tem de se ter em atenção alguns pormenores nos movimentos mecânicos do *DT*, para verificar se o tempo desta, será o mais correto, pois se não for, deve ser iniciada novamente a simulação.

Por outro lado, existiram problemas gerais nalgumas simulações, a nível mecânico dos elementos modelados. Em cada simulação executada, a adversidade maior encontrada, foi das peças nem sempre serem movimentadas corretamente, afetando o adequado funcionamento da sequência de montagem atribuída. Por vezes, foi necessário ajustar as medidas em milímetros de alguns componentes, como a base das peças, o comutador de peças, os pinos da roda e as próprias peças, para otimizar esta movimentação.

Também ocorre o facto, das peças não encaixarem bem nos pinos do posto, porque ficavam presas e não desciam por gravidade, bem como a entrarem umas por dentro das outras, devido a terem o corpo de colisão do tipo malha, muito idêntico. No primeiro caso, o fator convexo do corpo de colisão de cada peça utilizada, tem de ser ajustado. Isto permite que a peça seja melhor introduzida no posto, e tornar o corpo de colisão associado à peça, mais aberto ou mais fechado. Para o segundo caso, as peças podem ser apenas rodadas ou alterar também o fator convexo. Um facto curioso, é que a primeira situação descrita, também ocorre no *kit* real, como se pode verificar na Figura 92. A solução para o problema, poderia ser diminuir o diâmetro dos pinos, onde são colocadas as peças.

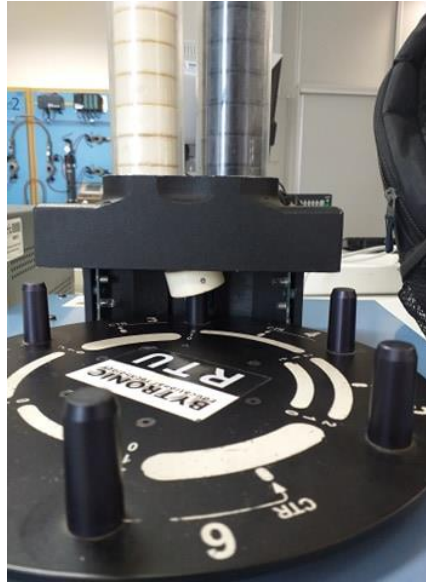


Figura 92 – Problema no kit real de colocação de peça no pino

Sendo a simulação incerta, o resultado obtido nem sempre era o desejado. No entanto, procedimentos foram elaborados, para que de uma forma geral, a simulação no *NX MCD*, com um controlo externo, tenha um comportamento razoável, de acordo com o processo real. Quanto às interligações estudadas, utilizando o *PLC* real, conclui-se que o melhor resultado obtido foi com as interligações por *TCP/IP* e por *OPC DA*, devido ao facto, de ser possível definir o tempo de atualização, menor que as outras. Com o protocolo *Profinet*, não foi possível obter sucesso. Com o *OPC UA*, devido ao tempo de amostragem e publicação do *Server* criado pelo *PLC*, não é possível obter uma resposta tão rápida na interligação. Por *OPC UA*, alguns elementos como sensores, tiveram de ser ligeiramente alterados, para tentar compensar esse atraso na interligação, sendo menos preciso que as outras. Para além da exploração do software para a realização destas interligações, também foram elaboradas técnicas com o auxílio de artigos, tutoriais e fóruns da *Siemens*[®] [93][94].

5. Conclusões

Neste projeto, foi desenvolvido um *DT* de um processo real existente, que tem como objetivo demonstrar o potencial deste, no conceito da Indústria 4.0. Foi possível demonstrar a importância da virtualização de processos, que tem impacto no comissionamento de um sistema real, possibilitando a verificação do seu funcionamento e a sua otimização. Para além disso, é possível adquirir dados e identificar melhorias ao longo do tempo. Uma empresa no setor industrial, pode usufruir da virtualização, no sentido de elaborar planeamentos de trabalho e manutenção, para tornar mais eficiente o funcionamento da empresa.

O trabalho foi desenvolvido na *Siemens Automation Academy*, tendo como ponto de partida, o objetivo de explorar o *software* e *hardware* da *Siemens*[®]. Nessa investigação, pretendeu-se criar um *DT*, de um processo real existente na academia. Foi usado o *software NX*, que permite realizar a modelação 3D e a estimulação do gémeo digital. Também foram utilizados *PLCs* reais, para interligar com o *DT* e efetuar o seu controlo, através do *software TIA Portal*. O *NX*, pode oferecer uma solução digital, na área da indústria, no que toca a processos reais que se pretendem implementar ou implementar otimizações nas tarefas a executar.

Neste trabalho, foram explorados vários protocolos de comunicação, que são possíveis aplicar com este *software*. Foi possível obter sucesso em todas as interligações efetuadas, desde o *PLCSIM*, *TCP/IP*, *OPC DA* e *OPC UA*, exceto com o protocolo *Profinet*. Talvez pelo facto, de ser relativamente recente, utilizar comunicação externa com este software, conclui-se que existe ainda pouca informação e documentação, sobre as interligações possíveis. Para futuros estudos sobre o *DT*, usando o *NX*, seria importante existir mais informação e tutoriais sobre as interligações.

No cômputo geral, o *DT*, é um conceito inovador que pode ser aplicado em vários sistemas do nosso dia-a-dia. Esta tecnologia permite obter uma cópia virtual de um processo físico, onde é possível obter informações, que nos permitem interagir com os processos, através de ajustes de parâmetros, tornando estes mais eficientes. Este projeto, pode dar asas ao desenvolvimento de *DTs*, correspondentes a outros processos físicos, e explorar outras vertentes da sua potencialidade, para atingir melhorias na parte física e digital.

O protocolo *OPC UA*, é um protocolo de comunicação com potencial, na Indústria 4.0, visto possibilitar a ligação entre várias redes de comunicação distintas, tendo como característica a

interoperabilidade. Pode ser aplicado como uma interface *standard*, entre um equipamento real e *DTs*. Como trabalho futuro, pode se explorar as potencialidades deste protocolo, relativas ao processo estudado, como desenvolver modelos de informação mais detalhados, e implementar armazenamento de dados do processo, em bases de dados locais ou em *Cloud*.

Quanto aos conhecimentos adquiridos, foi interessante elaborar a modelação 3D, visto o autor não ter bases sobre mecânica nem desenho. Foi desafiante, identificar as operações necessárias a elaborar, para obter o modelo 3D, do *DT* pretendido. Durante a realização deste projeto, foi cativante perceber a importância que este pode trazer para o ramo da indústria, principalmente no facto de simular um processo físico, de forma digital, antes de este ser implementado. Era interessante que mais projetos fossem desenvolvidos sobre o conceito do *DT*, para entender melhor o seu impacto na indústria, e divulgar este, para que as empresas tenham noção, da sua importância nos seus processos.

Por fim, este trabalho participou no prémio geração digital !19, realizado pela *Siemens*[®] Portugal, tendo ficado em primeiro lugar. Neste projeto, foi utilizado *software* da *Siemens*[®], e aplicados conceitos da Indústria 4.0, que são importantes no crescente aumento da tecnologia.

Referências Bibliográficas

- [1] Bernard Marr, What is Industry 4.0? Here's a super easy explanation for anyone, <https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/>, Forbes, consultado em maio 2019
- [2] Carlos Ribas, Bosch Portugal, Indústria 4.0, a quarta revolução industrial - <https://jornaleconomico.sapo.pt/noticias/nao-aprovar-industria-4-0-a-quarta-revolucao-industrial-182746>, 2017, Jornal Económico, consultado em junho 2019
- [3] Greyce N. Schroeder, Charles Steinmetz, Carlos E. Pereira, Danubia B. Espindola, Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange, DOI: 10.1016/j.ifacol.2016.11.115
- [4] Thomas H.-J. Uhlemann, Christian Lehmann, Rolf Steinhilper, The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0, DOI: 10.1016/j.procir.2016.11.152
- [5] Dr. Ulrich Rossgoderer, Digital Twin – Plan, Optimize and Visualize Your Factory, 2017
- [6] Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0, <https://www.i-scoop.eu/industry-4-0/>, consultado em Junho 2019
- [7] Roland Berger, Conceito de Indústria 4.0 – Siemens, 2016
- [8] Andreja Rojko, Industry 4.0 Concept: Background and Overview, DOI: 10.3991/ijim.v11i5.7072
- [9] Siewhui Chong, Guan-Ting Pan, Jitkai Chin, Pau Loke Show, Thomas Chung Kuang Yang, Chao-Ming Huang, Integration of 3D Printing and Industry 4.0 into Engineering Teaching, DOI: 10.3390/su10113960
- [10] Julie Dirksen, Dustin DiTommaso, Cindy Plunkett, Augmented and virtual reality for behaviour change, DOI: 10.13140/RG.2.2.23504.35842
- [11] Qinglin Qi, Fei Tao, Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison, DOI: 10.1109/ACCESS.2018.2793265
- [12] Adrienn Skrop, Industry 4.0 – Challenges in Industrial Artificial Intelligence, International scientific conference on tourism and security 4 december 2018

-
- [13] Kazi Masudul Alam, Abdulmotaleb El Saddik, C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems, DOI: 10.1109/ACCESS.2017.2657006
- [14] The Internet of Things (IoT) – essential IoT business guide, <https://www.i-scoop.eu/internet-of-things-guide/>, consultado em julho 2019
- [15] Paul-Eric Dossou, Development of a new framework for implementing industry 4.0 in companies, DOI: 10.1016/j.promfg.2020.01.072
- [16] João Ribau, Afinal, o que é a Indústria 4.0? - <https://visao.sapo.pt/opiniao/ponto-de-vista/2019-10-28-afinal-o-que-e-isto-da-industria-4-0/>, 2019, consultado em janeiro 2020
- [17] Klaus-Dieter Thoben, Thorsten Wuest, Stefan Alexander Wiesner, "Industrie 4.0" and Smart Manufacturing - A Review of Research Issues and Application Examples, DOI: 10.20965/ijat.2017.p0004
- [18] Hüseyin Şenkayas, Özden Gürsoy, Industry 4.0 applications and digitalization of lean production lines, Handle: RePEc:ora:journl:v:1:y:2018:i:1:p:124-131
- [19] Radoslaw Miśkiewicz, Radoslaw Wolniak, Practical application of the Industry 4.0 concept in a steel company, DOI: 10.3390/su12145776
- [20] Stephan Simons, Patrick Abé, Stephan Naser, Learning in the AutFab – the fully automated Industrie 4.0 learning factory of the University of Applied Sciences Darmstadt, DOI: 10.1016/j.promfg.2017.04.023
- [21] Ellen Kou, Achieving the Industrial Internet of Things through Virtualization - <https://www.5gtechnologyworld.com/achieving-the-industrial-internet-of-things-through-virtualization/>, 2018, consultado em julho 2019
- [22] Stefan Zimmermann, Industry 4.0 – how plant/asset virtualization will change asset planning and optimization - <https://atos.net/en/blog/industry-4-0-plant-asset-virtualization-will-change-asset-planning-optimization>, 2019, consultado em agosto 2019
- [23] Ole Hanseth, Philosophy and industrialized software development, Scandinavian Journal of Information Systems: Vol. 3 : Iss. 1 , Article 2
- [24] Empresa IMV, na qual o autor trabalha atualmente - <https://imvlda.com/>, consultado em agosto 2020

-
- [25] PHC Software - <https://www.phcsoftware.com/>, consultado em agosto 2019
- [26] Empresa Megatech na qual o autor trabalhou - <https://www.megatech-industries.com/pt/>, consultado em agosto 2019
- [27] What is HMI? - <https://www.inductiveautomation.com/resources/article/what-is-hmi>, 2018, consultado em agosto 2019
- [28] Dr. A. Selwin Mich Priyadharson, Prabhu Kumar Surarapu, PLC-HMI based automarion screen printing system, International Journal of Mechanical Engineering and Technology (IJMET), Volume 8, Issue 10, October 2017, pp. 101–107, Article ID: IJMET_08_10_013
- [29] Erik Aranburu, Ganix Lasa, Jon Kepa Gerrikagoitia, Maitane Mazmela, Case study of the Experience Capturer Evaluation Tool in the Design Process of an Industrial HMI, DOI: 10.3390/su12156228
- [30] What is SCADA? - <https://inductiveautomation.com/resources/article/what-is-SCADA>, 2018, consultado em agosto 2019
- [31] Projeto Secure Plug an work - https://www.plattform-i40.de/PI40/Redaktion/EN/Use-Cases/004-secure-plug-and-work-fraunhofer-iosb_EN/article-secure-plug-and-work-fraunhofer-iosb_EN.html, consultado em setembro 2020
- [32] Cesar Da Costa, Cleiton Rodrigues Mendes, Raphael Osaki, Industry 4.0 in automated production, DOI: 10.13140/RG.2.2.30849.15204
- [33] Judit Nagy, Judit Oláh, Edina Erdei, Domicián Máté, József Popp, The Role and Impact of Indutry 4.0 and the Internet of Things on the Business Strategy of the Value Chain – The Case of Hungary, DOI: 10.3390/su10103491
- [34] Dr. Michael Grieves, John Vickers, Origins of the Digital Twin Concept, DOI: 10.13140/RG.2.2.26367.61609
- [35] Aaron Parrott, Dr. Lane Warshaw, Deloitte, Industry 4.0 and the Digital Twin, 2017
- [36] Thomas H.-J. Uhlemann, Christoph Schock, Christian Lehmann, Stefan Freiberger, Rolf Steinhilper, The Digital Twin: Demonstrating the potential of real time data acquisition in production systems, DOI: 10.1016/j.promfg.2017.04.043
- [37] Benjamin Schleich, Nabil Anwer, Luc Mathieu, Sandro Wartzack, Shaping the Digital Twin for design and production engineering, DOI: 10.1016/j.cirp.2017.04.040

- [38] Fei Tao, Meng Zhang, Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing, DOI: 10.1109/ACCESS.2017.2756069
- [39] Siemens, Why digital twins will be the backbone of industry in the future, https://www.youtube.com/watch?v=ObGhB9CCHP8&t=22s&ab_channel=Siemens, 2019, consultado em setembro 2019
- [40] Ján Vachálek, Lukáš Bartalský, Oliver Rovný, Dana Šišmišová, The Digital Twin of an Industrial Production Line Within the Industry 4.0 Concept, DOI: 10.1109/PC.2017.7976223
- [41] Gloria Maria Alves Ney, Márcio Zamboti Fortes, Metodologia para processo integrado de comissionamento em unidades industriais, DOI: 10.20985/1980-5160.2017.v12n2.1189
- [42] Digital Twins and Virtual Commissioning in Industry 4.0 - <https://www.visualcomponents.com/insights/articles/digital-twins-and-virtual-commissioning-in-industry-4-0/>, 2019, consultado em setembro 2019
- [43] Syed Manzar Abbas Kazmi, Master's Thesis. Methodology for validating mechatronic Digital Twin, 2019, <http://urn.fi/URN:NBN:fi:tti-201905211696>
- [44] Joonas Taipalus, Thesis. 3D-Virtualization of a Conveyor Machine, 2015
- [45] Stephanie Demers, Praveen Gopalakrishnan, Latha Kant, A Generic Solution to Software-in-the-loop, DOI: 10.1109/MILCOM.2007.4455268
- [46] Karima Tebani, Brahum Mihoubi, Graham Mehdi, M. Gaham, B. Bouzounia, Hardware-in-the-loop Simulated for Validating PLC Programs, 2015, Conference: CIAM'15 At: Oran Algérie
- [47] Mohamed Fasil Syed Ahamed, Girma Tewolde, Jaerock Kwon, Software-in-the-loop Modeling and Simulation Framework for Autonomous Vehicles, DOI: 10.1109/EIT.2018.8500101
- [48] Daynier Rolando, Delgado Sobrino, Roman Ružarovský, Radovan Holubek, Karol Velišek, Into the early steps of Virtual Commissioning in Tecnomatix Plant Simulation using S7-PLCSIM Advanced and STEP 7 TIA Portal, DOI: 10.1051/mateconf/201929902005

-
- [49] Jonis Kiesbye, David Messmann, Maximilian Preisinger, Gonzalo Reina, Daniel Nagy, Florian Schummer, Martin Mostad, Tejas Kale, Martin Langer, Hardware-in-the-loop and Software-in-the-loop Testing of the MOVE-II CubeSat, DOI: 10.3390/aerospace61201
- [50] Harpreet Singh Bedi, Shekhar Verma, Rk Sharma, Balpreet singh, The Concept of Programmable Logic Controllers and its role in Automation, DOI: 10.17148/IJARCCE.2015.46120
- [51] Ephrem Ryan Alphonsus, Mohammad Omar Abdullah, A review on the applications of programmable logic controllers (PLCs), DOI: 10.1016/j.rser.2016.01.025
- [52] Denis Bobylev, Thesis, Comparison of 3d modeling software, 2017
- [53] Siemens – TIA Portal,
<https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>, consultado em novembro 2020
- [54] Rainer Stark, Simon Kind, Sebastian Neumeyer, Innovations in digital modelling for next generation manufacturing system design, DOI: 10.1016/j.cirp.2017.04.045
- [55] Siemens NX - <https://www.plm.automation.siemens.com/global/en/products/nx/>, consultado em outubro 2019
- [56] Mechatronics Concept Design -
<https://www.plm.automation.siemens.com/global/en/products/mechanical-design/mechatronic-concept-design.html>, consultado em outubro 2019
- [57] Site Real Games software Factory I/O - <https://realgames.co/>, consultado em novembro 2020
- [58] João Vieira, Tiago Neves, Projeto CoMoPro – Controlo e Monitorização de Processos, <https://siemensautomationacademy.ipleiria.pt/projetos/comopro-20152016/>, 2016
- [59] Site ABB® software RobotStudio - <https://new.abb.com/products/robotics/robotstudio>, consultado em novembro 2020
- [60] André Martins, Hugo Costelha, Shop Floor Virtualization and Industry 4.0, DOI: 10.1109/ICARSC.2019.8733657
- [61] Site do software Emulate 3D - <https://www.demo3d.com/>, consultado em novembro 2020

- [62] Ian McGregor, Introduction to Emulate3D – Emulators, Simulation, and Demonstration, 2012, Conference: Proceedings of the Winter Simulation Conference
- [63] Site do software Delmia - <https://www.3ds.com/products-services/delmia/>, consultado em novembro 2020
- [64] Waldemar Malopolski, A method for the automatic creation of bidirectional transportation paths for AGVs in Delmia Quest, DOI: 10.4467/2353737XCT.19.056.10580
- [65] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche, Design, modelling, simulation and integration of cyber physical systems: Methods and applications, DOI: 10.1016/j.compind.2016.05.006
- [66] Fredrik Danielsson, Philip Moore, Patric Eriksson, Validation, off-line programming and optimisation of industrial control logic, DOI: 10.1016/S0957-4158(02)00030-2
- [67] Chi G. Lee, Sang C.Park, Survey on the Virtual Commissioning of manufacturing systems, DOI: 10.7315/JCDE.2014.021
- [68] Peter Hoffmann, Talal M.A. Maksoud, Virtual Commissioning on manufacturing systems a review and new approaches for simplification, DOI: 10.7148/2010-0175-0181
- [69] PLC Communications – Coming of Age – <https://library.automationdirect.com/plc-communications-coming-of-age/>, consultado em novembro 2019
- [70] What is OPC? - <https://opcfoundation.org/about/what-is-opc/>, consultado em novembro 2019
- [71] Guia Definitivo: Entenda tudo sobre o que é OPC UA e como isso pode impactar a sua indústria - <https://www.logiquesistemas.com.br/blog/opc-ua/>, consultado em novembro 2019
- [72] OPC and OPC UA explained - <https://www.novotek.com/uk/solutions/keeware-communication-platform/opc-and-opc-ua-explained/>, consultado em novembro 2019
- [73] Julius Pfrommer, Andreas Ebner, Siddharth Ravikumar, Bhagath Karunakaran, Open Source OPC UA PubSub over TSN for Realtime Industrial Communication, DOI: 10.1109/ETFA.2018.8502479

-
- [74] Iman Ayatollahi, Burkhard Kittl, Florian Pauker, M. Hackhofer, Prototype OPC UA Server for Remote Control of Machine Tools, 2013, International Conference on Innovative Technologies, IN-TECH 2013, Budapest, 10. - 12.09.2013
 - [75] Gustavo Caiza, Alex Nuñez, Carlos A. Garcia. Marcelo V. Garcia, Human Machine Interfaces on Open Source Web-Platform and OPC UA, DOI: 10.1016/j.promfg.2020.02.089
 - [76] Xun Ye, Seung Ho Hong, Yuemin Ding, Implementation of A Production-Control System using Integrated AutomationML and OPC UA, DOI: 10.1109/METROI4.2018.8428310
 - [77] Ivan Carlos Franco, Jones Erni Schmitz, Ana Maria Frattini Fileti, Flávio Vasconcelos da Silva, Utilização do protocolo de comunicação OLE for Process Control em processos industriais, DOI: 10.5585/exacta.v8i3.2431
 - [78] Timo Oksanen, Raimo Linkolehto, Ilkka Seilonen, Adapting an industrial automation protocol to remote monitoring of mobile agricultural machinery: a combine harvester with IoT, DOI: 10.1016/j.ifacol.2016.10.024
 - [79] Bytronic, Rotary Transfer Unit RTU - <http://www.bytronic.net/wp-content/uploads/2015/05/RTU-PL0111.pdf>, consultado em junho 2019
 - [80] Mechatronics Concept Designer guide - https://www.academia.edu/20062592/NX_10_-_Mechatronics_Concept_Design_-_Gu%C3%ADa, consultado em novembro 2019
 - [81] Workstation Configurations For Siemens PLM NX - <https://www.intel.com/content/dam/www/public/us/en/documents/guides/siemens-workstation-configuration-guide.pdf>, consultado em novembro 2019
 - [82] Siemens Mechatronics Concept Designer Quick start and User Manual – http://www2.me.rochester.edu/courses/ME204/nx_help/en_US/graphics/fileLibrary/nx/mechtronics/MCD_Quick_Start.pdf, consultado em novembro 2019
 - [83] NX Mechatronics Concept Designer – Tutorial 2 – Optimizations and modification of the simulation, https://www.youtube.com/watch?v=JyGV04uviqg&ab_channel=CRAatHDA, 2017, consultado em outubro 2019

- [84] Simatic S7-1200, Siemens -
https://www.atec.pt/images/stories/eventos/2010/sales_presentation_pt.pdf, consultado em dezembro 2019
- [85] Simatic S7-1500 mais TIA Portal, Siemens -
https://c4b.gss.siemens.com/resources/images/articles/e20001-a930-p210-v1-7900_dffa-b10011-00-7900.pdf, consultado em dezembro 2019
- [86] Manuel Goster, Bachelor Thesis, Concept and Implementation of a Factory Simulation, Ulm University, Germany, 2017
- [87] Programming guideline for S7-1200/S7-1500 -
http://media.automation24.com/manual/fr/90885040_81318674_Programming_guideline_DOC_v15_en.pdf, consultado em dezembro 2019
- [88] Open user communication with TSEND_C and TRCV_C -
https://cache.industry.siemens.com/dl/files/808/67196808/att_108115/v3/net_S7-1200_isoontcp_en.pdf, consultado em janeiro 2020
- [89] Creating and assigning parameters to SIMATIC PC stations -
http://www.plccenter.cn/Siemens_Step7/Erzeugen_und_parametrieren_von_SIMATIC_PC_Stationen.htm, consultado em janeiro 2020
- [90] I-device function in standard PN communication -
https://cache.industry.siemens.com/dl/files/798/109478798/att_856108/v4/109478798_config_iddevice_standard_DOCU_V1d0_en.pdf, consultado em fevereiro 2020
- [91] Commisioning PC stations – Manual and Quick start -
<https://www.manualslib.com/manual/524488/Siemens-Simatic-Net.html?page=153>, consultado em fevereiro 2020
- [92] Connecting a PC station to an S7-1200 using OPC -
https://cache.industry.siemens.com/dl/files/679/39960679/att_117996/v3/39960679_S7-1200_opc_simatic-net_tia-portal_e.pdf, consultado em março 2020
- [93] Fórum Siemens PLM Community - <https://community.sw.siemens.com/s/>, consultado em Junho 2020
- [94] Fórum Industry Support Siemens - <https://support.industry.siemens.com/cs/start>, consultado em Junho 2020

Anexos



Tia
Portal_FC_Grafcet.pdf